

1998

VLSI library cells for cellular neural network applications.

Anu. Mathew
University of Windsor

Follow this and additional works at: <http://scholar.uwindsor.ca/etd>

Recommended Citation

Mathew, Anu., "VLSI library cells for cellular neural network applications." (1998). *Electronic Theses and Dissertations*. Paper 4538.

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

**Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600**

UMI[®]

VLSI Library Cells for Cellular Neural Network Applications

by

Anu Mathew

A Thesis

Submitted to the Faculty of Graduate Studies and Research through the

Department of Electrical Engineering in Partial Fulfillment

of the Requirements for the Degree of

Master of Applied Science

at the

University of Windsor

Windsor, Ontario.

December 1997.



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-52470-1

Canada

© 1997 by Anu Mathew

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording or any information storage and retrieval system, without the prior written permission of the author.

**Dedicated to
The Almighty God**

Acknowledgements

I would like to express my sincere thanks and appreciation to my supervisor **Dr. G.A Jullien** for his invaluable support and guidance throughout the progress of this thesis. I would also like to express my profound gratitude towards my committee members **Dr. M Ahmadi** and **Dr. A Jaekel** for their constructive suggestions and encouragement.

Thanks to my many colleagues, particularly **Dr. Binqiao Li**, **Saeid Sadeghi-Emamchaie**, **Dr. Vassil Dimitrov**, **Roberto Muscedere** and **Marjan Shahkarami** who really contributed towards the success of this thesis. Moreover, I take this opportunity to express my deep gratitude to all the VLSI group members of the University of Windsor.

My special thanks are also due to my parents and family members for their advice and prayers. Finally, my very special thanks to my wife **Lisbeth** for her indefatigable support and encouragement.

Abstract

Cellular Neural Networks (CNNs) are massively parallel nonlinear locally connected analog cells; they are often targeted for use in image processing applications. An analog signal processor which (in majority of cases) does not require A/D conversion, occupies only a fraction of the area occupied by its digital counterpart. In image processing, it is possible to integrate an analog processor with each signal source (pixel) without leading to impractically low signal-source (or pixel) density. This makes possible the parallel loading or injecting of the input signal into the analog processor. Thus, sequential sampling at the output voltage (or current) of each signal-source is eliminated and a high degree of parallelism in signal processing is easily achievable. With the use of modified and creative algorithms, Cellular Neural Networks may also be used for digital arithmetic operations. For the same speed, these analog processors have lower slew rates compared to their digital counterparts which, in turn, leads to a lower generated noise. Using a Cellular Neural Network architecture, this thesis focuses on silicon implementation and experimental characterization of the building blocks for image processing and binary arithmetic applications using the MOSIS 0.5 μ m technology. These library cells are verified functionally at the layout level by conducting DRC, LVS, and post layout simulations. Sensitivity analysis is also carried out on a basic CNN cell in order to determine its tolerance with respect to expected variations in process parameters.

Table of Contents

Chapter 1 INTRODUCTION	1
1.1 Definition	1
1.2 General Architecture	2
1.3 Functional Block Diagram of a Cell	3
1.3.1 The Circuit Equations of a Cell	7
1.3.2 Stability of CNN	9
1.4 CNN with Nonlinear and Delay type Templates	9
1.5 Outline of the Dissertation	10
Chapter 2 CNN APPLICATIONS	12
2.1 Image Processing	12
2.1.1 CNN Architecture with Programmable Templates	13
2.2 Binary Arithmetic Operations	15
2.2.1 Binary Addition	15
2.2.2 Rule 1:	16
2.2.3 Rule 2:	16
2.3 Self- Feedback Template	17
2.4 Equivalent Circuit Diagram of a Cell	20
Chapter 3 VLSI IMPLEMENTATION	22
3.1 Nonlinear V/I Converter	22
3.1.1 Programmable Current Scaling Circuit	26
3.2 Tunable Linear V/I Converter	30
3.3 Cell Resistor and State Capacitor	36
3.3.1 State capacitor	36
3.4 VLSI Library Cells for Binary Addition	40
3.4.1 Absolute Value Function Circuit	40
3.4.2 Current Comparator	44
3.4.3 Hysteresis Circuit	47
3.4.4 Current Adder and Subtractor Circuits	51
3.5 Basic CNN Cell	55
3.6 Verification of Rule 2:	59
Chapter 4 SENSITIVITY ANALYSIS	63
4.1 Mismatches in MOS Transistors	63
4.1.1 Threshold Voltage Mismatch	64
4.1.2 Conductance Term Mismatch	65
4.2 Sensitivity Analysis	66
4.2.1 Sensitivity analysis of the basic CNN cell	66
4.2.2 Observations	72
4.3 Summary	72
Chapter 5 CONCLUSIONS	73
REFERENCES	76
Appendix A SPICE netlist of OTA circuit	80
Appendix B SPICE netlist of linear V/I converter	81

Appendix C SPICE netlist of programmable V/I converter.....	83
Appendix D SPICE netlist of absolute value function circuit.....	87
Appendix E SPICE netlist of hysteresis circuit.....	90
Appendix F SPICE netlist of current adder and subtractor.....	92
Appendix G SPICE netlist of a basic CNN cell.....	94
Appendix H SPICE netlist of Rule 2 verification.....	97
Vita.....	105

List of Figures

Figure 1.1	A two dimensional 4×4 CNN. The squares are the cells and the link between the cells indicate their interaction	2
Figure 1.2	The neighbourhood of cell for $r=1$ and $r=2$ respectively.....	3
Figure 1.3	Functional block diagram of a cell.....	4
Figure 1.4	Equivalent circuit diagram of a cell	5
Figure 1.5	The characteristic of the nonlinear voltage controlled current source	6
Figure 1.6	Interaction between two neighbour cells C and D	7
Figure 2.1	Electronic equivalent circuit of a cell with programmable template.....	14
Figure 2.2	Application of rule 1	16
Figure 2.3	Application of rule 2	16
Figure 2.4	Hysteresis characteristic of	18
Figure 2.5	Equivalent circuit diagram	20
Figure 2.6	Detailed equivalent diagram of a cell.....	21
Figure 3.1	Simple transconductance amplifier	24
Figure 3.2	DC response of OTA	25
Figure 3.3	Current scaling circuit for nonlinear V/I converter	27
Figure 3.4	Cascading of nonlinear V/I converter with a current scaling circuit.....	28
Figure 3.5	Output response of the scaling circuit for different <i>V</i> value Voltages.....	29
Figure 3.6	Basic cell of linear transconductance circuit	32
Figure 3.7	Linear tunable transconductance circuit	33
Figure 3.8	Modified linear tunable transconductance circuit	34
Figure 3.9	Output response of a linear transconductance circuit	35
Figure 3.10	Cell resistor circuit.....	38
Figure 3.11	DC. response of the cell resistor	39
Figure 3.12	Absolute value function circuit	41
Figure 3.13	DC response of the absolute value circuit.....	42
Figure 3.14	Layout of the absolute value circuit.....	43
Figure 3.15	Current comparator	44
Figure 3.16	DC response of the current comparator	45
Figure 3.17	Layout of current comparator	46
Figure 3.18	Hysteresis function circuit	48
Figure 3.19	DC. response of the hysteresis circuit.....	49
Figure 3.20	Transient response of the circuit	49
Figure 3.21	Layout of the hysteresis circuit.....	50
Figure 3.22	Current adder	51
Figure 3.23	Current subtractor	52
Figure 3.24	DC response of current subtractor	53
Figure 3.25	DC response of current adder	54
Figure 3.26	Basic cell circuit.....	56

Figure 3.27	Transient response when initial voltage is 4V	57
Figure 3.28	Transient response when initial voltage is 0.5V	57
Figure 3.29	Layout of the basic CNN cell.....	58
Figure 3.30	Verification of rule 2	60
Figure 3.31	Output of the cell N1&N2 before switching.....	61
Figure 3.32	Output after switching.....	62
Figure 4.1	Varying V_t of NMOS	68
Figure 4.2	Varying V_t of both NMOS and PMOS.....	69
Figure 4.3	Varying K_p of NMOS	69
Figure 4.4	Varying K_p of both NMOS and PMOS.....	70
Figure 4.5	Varying V_t and K_p of both NMOS and PMOS	70

List of Tables

Table 4.1.	Values of V_t and K_p for a typical CMOSIS5 process	67
Table 4.2.	Measured output current from different combinations	68
Table 4.3.	Influence of V_t	71
Table 4.4.	Influence of K_p	71
Table 4.5.	Influence of V_t and K_p	71

Chapter 1

INTRODUCTION

Alternative computational architectures, based on some aspects of neurobiology and adapted to integrated circuits[1][2], have been explored over the past 2 decades. A recently introduced locally connected architecture, referred to as Cellular Neural Networks (CNNs) [3], has recently emerged and is proving to be a powerful new computational paradigm. The CNN is a locally connected non-linear analog circuit architecture adaptable to a variety of array computations. Array computations can be defined as the parallel execution of complex operations on a large number of processors placed on a geometrically regular grid. CNNs have recently been targeted for several image processing application areas.

1.1 Definition

Cellular Neural Networks are n -dimensional arrays of identical dynamical systems, called cells, which satisfy the following two properties[4]:

1. Most interactions are local within a finite radius r .
2. All state variables are continuously valued signals.

A template specifies the interaction between each cell and all its neighbouring cells in terms of their input, state and output variables.

1.2 General Architecture

The basic CNN is a continuous time network in the form of a $M \times N$ rectangular grid array arranged in M rows and N columns. The basic unit of the CNN, called a cell, corresponds to an element of the array. Notice that the geometry of the array need not be rectangular and can, in fact, be other configurations such as triangles or hexagons. Theoretically, we can define a CNN to be of any dimension; however, in this dissertation we will concentrate on the 2-dimensional case. A typical example is shown in Figure 1.1.

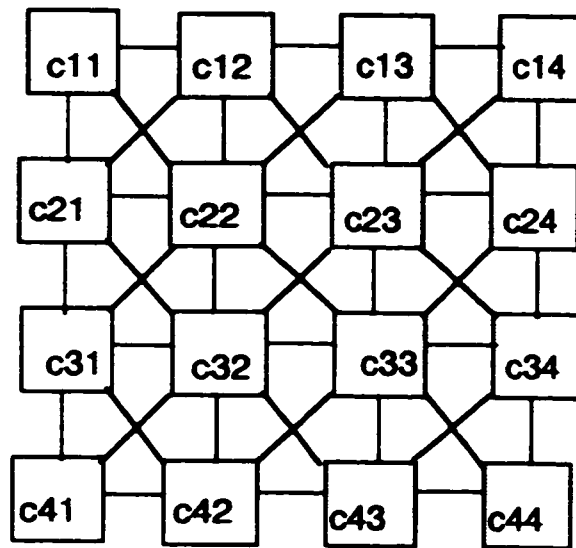


Figure 1.1 A two dimensional 4x4 CNN. The squares are the cells and the link between the cells indicate their interaction

For the $M \times N$ cells arranged in M -rows by N -columns, the r -th neighbourhood cells $N_r(i, j)$ of a cell $C(i, j)$, $1 \leq i \leq M$ and $1 \leq j \leq N$, are defined as the cells $C(k, l)$, $1 \leq k \leq M$ and $1 \leq l \leq N$, for which $|k - i| \leq r$ and $|l - j| \leq r$ such that:

$$N_r(i, j) = \{C_{k,l}; \max(|k-i|, |l-j|) = r(\text{integer})\} \quad (1.1)$$

Figure 1.2 shows 2-neighbourhoods of the same cell (located at the center and shown shaded) with $r=1$ and 2 respectively. With $r=1$ we have a 3×3 neighbourhood and with $r=2$ we have a 5×5 neighbourhood. These neighbourhood structures have been used in image processing applications described in CNN classical literatures[3][4].

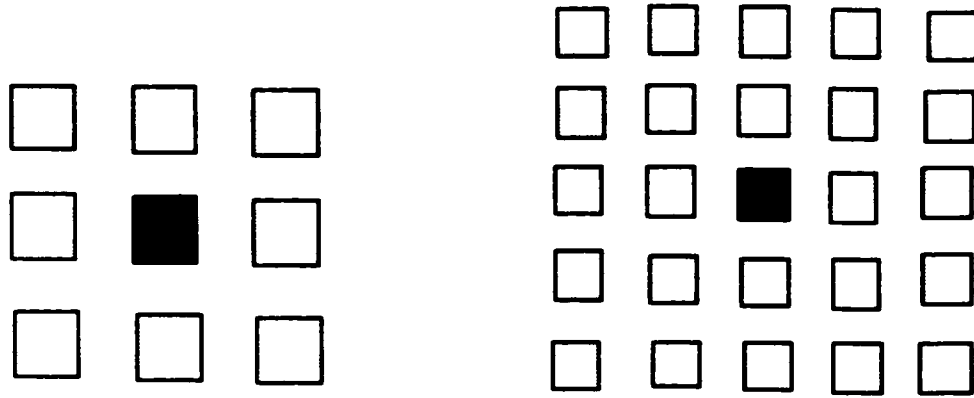


Figure 1.2 The neighbourhood of cell $C(i, j)$ for $r=1$ and $r=2$ respectively

1.3 Functional Block Diagram of a Cell

The cell $C(i,j)$ has direct interconnections with neighbours $N_r(i, j)$ through two kinds of weights, i.e. the feedback weights $A(i,j;k,l)$ & $A(k,l;i,j)$ and feedforward weights $B(i,j;k,l)$ & $B(k,l;i,j)$, where the index pair $(k,l;i,j)$ represents the direction of signal from cell $C(i,l)$ to cell $C(k,l)$. The cell $C(i,j)$ communicates directly with its neighbourhood cells $C(k, l) \in N_r(i, j)$. Since the cells $C(k,l)$ have their own neighbourhood cells, $C(i,j)$ also communicates with all other cells indirectly. The functional block diagram of a cell is shown in Figure 1.3[5]. The external input to the cell is denoted by $V_{uij}(t)$ and typically assumed to be constant over an operational interval $0 \leq t \leq T$. The input is connected to $N_r(i, j)$ through the feedforward weights $B(i,j;k,l)$. The output of the cell, denoted by

$V_{yij}(t)$, is coupled to the neighbourhood cells $C(k,l)$ through the feedback weights $A(i,j;k,l)$. Therefore, the input signals consist of the weighted sum of feedforward inputs and weighted sum of feedback inputs. In addition, a constant bias term is added to the cell. If the weights represent the transconductance values among the cells, the total input current to the cell is given by Eqn. (1.2)

$$I_{xij}(t) = \sum_{C(k,l) \in N_r(i,j)} A(ij;kl)V_{ykl}(t) + \sum_{C(k,l) \in N_r(i,j)} B(ij;kl)V_{ukl}(t) + I_b \quad (1.2)$$

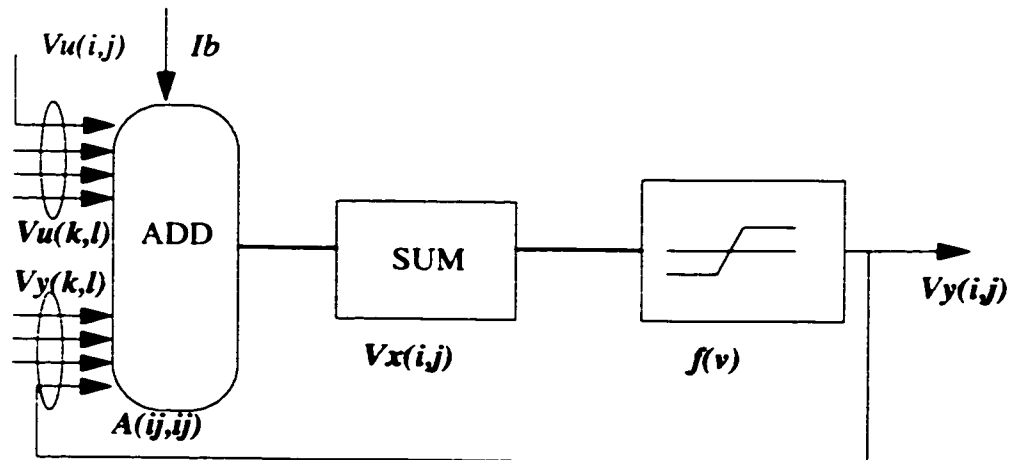


Figure 1.3 Functional block diagram of a cell

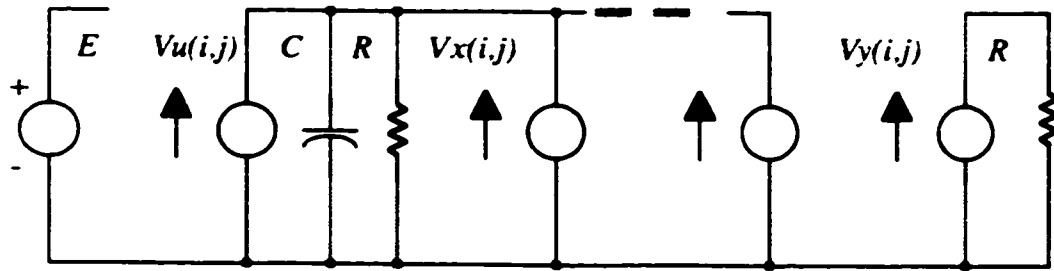


Figure 1.4 Equivalent circuit diagram of a cell

The equivalent circuit diagram of a cell $C(i,j)$ is shown in Figure 1.4, where R and C are the equivalent resistance and capacitance respectively. All inputs are represented by dependent current sources and summed at the state node. Due to the capacitance C and resistance R , the state voltage, V_{xij} , is established at the summing node and satisfies a set of differential equations. Referring to Figure 1.4, the suffices u , x , and y denote the input, state, and output respectively. Thus the node voltages of the cell $C(i,j)$ are the state voltage V_{xij} , the input voltage V_{uij} , and the output voltage V_{yij} . Observe from Figure 1.4 that each cell $C(i,j)$ contains one independent voltage source E , one independent current source I , one linear capacitor C , two linear resistors and at most $2m$ linear voltage controlled current sources, where m is equal to the number of neighbour cells. The linear voltage controlled current sources $I_{xy}(ij;kl)$ and $I_{xu}(ij;kl)$ have the following characteristics:

$$I_{xy}(ij;kl) = A(ij;kl)V_{ykl} \quad \text{and} \quad (1.3)$$

$$I_{xu}(ij;kl) = B(ij;kl)V_{ukl} \quad (1.4)$$

for all $C(k,l) \in N_r(i,j)$, where A & B are known as template coefficients. The only nonlinear element in each cell is a piecewise-linear voltage controlled current source

$$I_{yx} = \left(\frac{1}{R} \right) f(V_{xij}) \quad \text{with characteristic } f(.) \text{ as shown in Figure 1.5.}$$

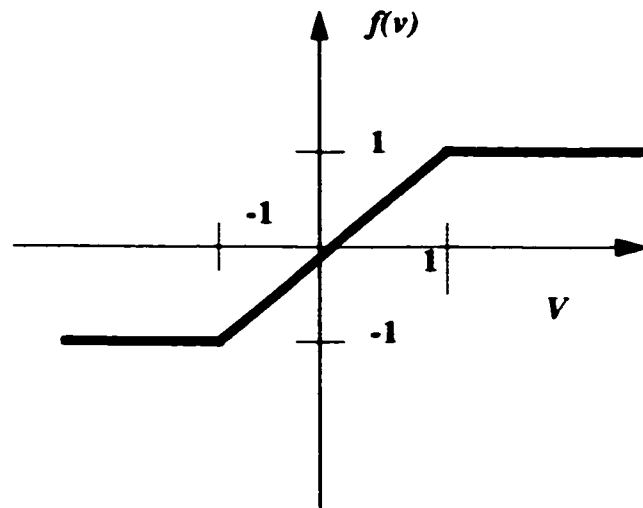


Figure 1.5 The characteristic of the nonlinear voltage controlled current source

To illustrate the interaction between neighbouring cells, consider a CNN with two cells (cell C and cell D) as shown in Figure 1.6[6]. Each cell (say cell C) receives external signals through its input voltage V_{uC} and has an independent current source I_c . Also, as indicated by the transconductance, the state voltage of a given cell V_{xC} is influenced not only by its own input V_{uC} , bias current I_c and its output feedback $AC_c V_{yC}$, but also by the input and output voltages of the neighbouring cell D. The output feedback effect depends on the interactive parameter A , which is referred to as feedback operator; the input control effect depends on the interactive parameter B , which is referred to as feedforward operator. Here AC_d indicates the direction of the signal from cell C to cell D and AD_c indicates interaction from cell D to cell C and so on.

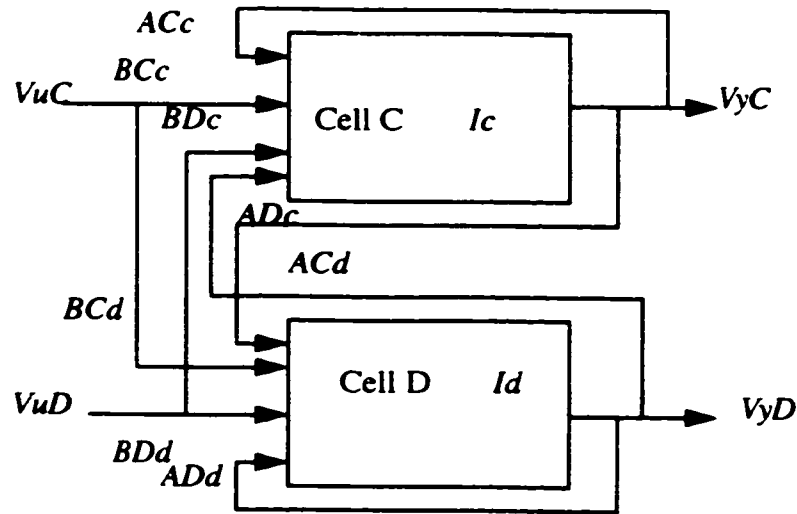


Figure 1.6 Interaction between two neighbour cells C and D

1.3.1 The Circuit Equations of a Cell

Applying Kirchoff's current and voltage laws (KCL and KVL) to the equivalent circuit of Figure 1.4, the circuit equations of a cell are easily derived as follows:

State Equation:

$$\begin{aligned}
 C \frac{dV_{xij}(t)}{dt} = & -\frac{1}{R} V_{xij}(t) + \sum_{C(k,l) \in N_r(i,j)} A(ij;kl) V_{ykl}(t) \\
 & + \sum_{C(k,l) \in N_r(i,j)} B(ij;kl) V_{ukl}(t) + I
 \end{aligned} \tag{1.5}$$

Output equation:

$$V_{yij}(t) = \frac{1}{2} (|V_{xij}(t) + 1| - |V_{xij}(t) - 1|) \quad \text{for } 1 \leq i \leq M \text{ and } 1 \leq j \leq N \tag{1.6}$$

Input Equation:

$$V_{uij}(t) = E_{ij} \quad \text{for } 1 \leq i \leq M \text{ and } 1 \leq j \leq N \quad (1.7)$$

Constraint Conditions:

$$|V_{xij}(0)| \leq 1 \quad \text{for } 1 \leq i \leq M \text{ and } 1 \leq j \leq N \quad (1.8)$$

$$|V_{uij}| \leq 1 \quad \text{for } 1 \leq i \leq M \text{ and } 1 \leq j \leq N \quad (1.9)$$

Parameter Assumptions:

$$A(ij;kl)=A(kl;ij) \quad 1 \leq i, k \leq M; 1 \leq j, l \leq N \quad (1.10)$$

$$C > 0, R > 0 \quad (1.11)$$

In order to guarantee the validity of the above circuit equations, the following theorems are proved which, in turn, can be used to determine the dynamic range of all node voltages in the network.

1. The state voltage V_{xij} of each cell in the CNN is bounded for all time $t > 0$ and the bound, V_{max} , can be computed by the following formula for any CNN:

$$V_{max} = 1 + R \times I + R \max_{1 \leq i \leq M, 1 \leq j \leq N} \left[\sum_{C(k,l) \in N_r(i,j)} (|A(ij;kl)| + |B(ij;kl)|) \right] \quad (1.12)$$

2. To ensure that the circuit will not oscillate or become chaotic, we must have

$$\lim_{t \rightarrow \infty} V_{xij}(t) = \text{constant, for } (1 \leq i \leq M) \text{ and } (1 \leq j \leq N) \quad (1.13)$$

3. To guarantee binary value outputs, the circuit parameters must satisfy

$$A(ij;ij) > \frac{1}{R} \quad (1.14)$$

1.3.2 Stability of a CNN array

Each cell of the CNN must settle at a stable equilibrium point after the transient has decayed to zero. Moreover, the magnitude of all stable equilibrium points is greater than 1. In other words, we have the following properties:

$$\lim_{t \rightarrow \infty} |V_{xij}(t)| > 1 \quad \text{for } 1 \leq i \leq M \text{ and } 1 \leq j \leq N \quad (1.15)$$

and

$$\lim_{t \rightarrow \infty} V_{yij}(t) = \pm 1 \quad \text{for } 1 \leq i \leq M \text{ and } 1 \leq j \leq N \quad (1.16)$$

1.4 CNN with Nonlinear and Delay type Templates

By introducing very simple nonlinear and delay type templates, the CNN becomes a powerful framework for general analog array dynamics[6]. Here, instead of the two linear controlled sources, defined by $A(ij;kl)*V_{ykl}$ and $B(ij;kl)*V_{ukl}$ associated with a typical cell $C(i,j)$ and a typical interacting cell $C(k,l)$, we allow nonlinear and delayed controlled sources defined by

$$\hat{A}(ij;kl)(V_{ykl}, V_{yij}) + A^T(ij;kl)V_{ykl}(t - \tau) \quad \text{and} \quad (1.17)$$

$$\hat{B}(ij;kl)(V_{ukl}, V_{uij}) + B^T(ij;kl)V_{ukl}(t - \tau) \quad (1.18)$$

where \hat{A} , \hat{B} and A^T , B^T are associated with the nonlinear and delay type templates respectively. Thus instead of having a linear voltage controlled current source in the A and B templates, we now have a nonlinear and/or delay type voltage controlled current source.

Notice that we also allow the output function to have wider range output voltages so that the saturation voltage is $\pm K$ instead of ± 1 . Further, the nonlinearity in the templates is a function of at most two variables, namely the output voltage of a cell $C(i,j)$ and that of a neighbour $C(k,l)$.

1.5 Outline of the Dissertation

The entire work has been developed through five chapters. An outline of the work reported in these chapters is described below

Chapter 1 gives a definition and general architecture of the CNN, functional block diagram, equivalent circuit and circuit equations of the cell, CNNs with non-linear and delay type templates are also presented along with a broad outline of the presented work.

Chapter 2 is devoted to the application of CNNs for image processing and binary arithmetic operations. In this chapter, the electronic equivalent circuit for image processing applications with a programming template is introduced. For binary arithmetic applications, the algorithm for the binary addition with their constraint conditions is included.

Chapter 3 is devoted to the VLSI implementation of the CNN applications. This includes building blocks for image processing applications, building blocks for binary arithmetic operations, and the basic CNN cell. All the building blocks have had their corresponding layouts and their functions verified using DRC, LVS, and post layout simulations using the Cadence design tool suite and HSPICE.

Chapter 4 presents results from a sensitivity analysis on the basic CNN cell. This is carried out by applying variations to process parameters of the target CMOS technology.

Finally, chapter 5 highlights the detailed conclusion of the entire work. Netlist of all the circuits designed in this research project are available in the Appendices.

Chapter2

CNN

APPLICATIONS

2.1 Image Processing

The CNN architecture is a uniform two dimensional array of locally interconnected analog cells which operate in continuous time. The interconnection strengths are spatially invariant and are organized in templates, which control the function of the network. Since the network contains no adaptive behavior or learning, the network can be better described as a nonlinear dynamic analog grid[7]. All cells operate in parallel and in continuous time. Notice that although the cells are locally connected, the network is able to perform global operations (e.g. image hole-filling)[8]. This is due to the dynamic nature of the network, information can propagate through the network and global properties of the 2-D data can be extracted.

The cell integrates the signal from its neighbours to calculate its state evolution and it sends an outgoing signal proportional to its output and input to its neighbours. The weights or scaling factors A and B for the outgoing signals are organized here in 3×3 matrices, called templates. The weights for the output related signals are collected in the A template and the scaling factors for the input related signals are in the B template. In every cell there is a constant

bias, contained in the I template. The three templates are identical for all cells in the grid and define the operation and function of the network.

Depending on the B template, the input image is loaded as either the inputs to the cells or as their initial states. After initialization, the states evolve to an equilibrium condition, and the final states are the output of the network. The speed of the operation is proportional to the cell time constant. This speed of the network depends on the ratio of the integrating capacitor to the transconductance of the nonlinear element $\left(\frac{C}{g_m}\right)$; hence, the convergence time is reduced by using small integrating capacitors and increasing g_m . It is well known that the transconductance gain of a CMOS transistor is proportional to the square root of both bias current and the gate dimensions. Therefore, there is a trade-off between speed of the network, silicon area and power consumption.

2.1.1 CNN Architecture with Programmable Templates

Figure 2.1 shows the electronic equivalent of a cell with programming templates[9][10]. Here programmability is defined as the ability to alter the value of the templates. By doing so, the network is able to implement many different processing algorithms. The transformation of the mathematical model for the cell behavior into an electronic circuit requires the choice of signal representations and of circuitry to perform the different functions. On a VLSI chip computation is performed by integrating current through capacitors[11][12][13]. The state X_i of the cell is then represented by the voltage across the state capacitor C . If the signals from the neighbours and the feedback are currents, they are easily summed with high precision through KCL. The input U_i of the cell remains constant during the cell evolution and can be stored as charge on the capacitor C_{input} (basically a dynamic analog memory), since the speed of the network is high. The loss term is realized with a linear cell resistor R in parallel with the state capacitor C . The generation of the output current signals requires a voltage to current conversion and a current scaling operation[14][15][16]. The A template signals are proportional to the cell

output Y_i . The nonlinear V/I converter converts the cell state voltage X_i to the cell output current Y_i using the nonlinearity $f(\cdot)$ of Figure 1.5. It has an input range specification equal to the state voltage swing. For the B template signals, the V/I converter is linear for the unit range $[-1,1]$ and its output current is proportional to the input U_i . The I template is a current source [17][18][19].

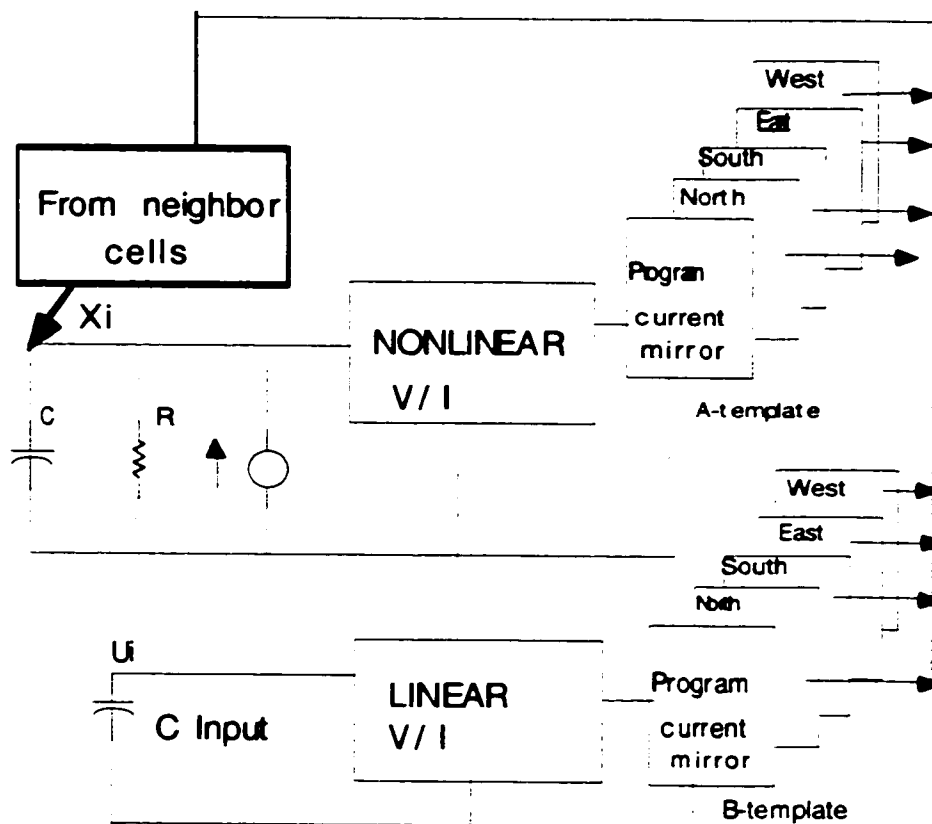


Figure 2.1 Electronic equivalent circuit of a cell with programmable template

2.2 Binary Arithmetic Operations

Cellular neural networks can be used to implement binary arithmetic arrays[20]. These arithmetic arrays are targeted at low speed/low noise applications where continuous power / speed trade-offs and lower slew rate during transition are the advantages. Here a CNN array using a simple nonlinear feedback template with hysteresis for each node can perform arbitrary length binary addition with good performance in terms of stability and robustness.

The state equation of a cell $C(i,j)$ for this application is

$$C \frac{d}{dt} X_{ij}(t) = -\frac{1}{R} X_{ij}(t) + \sum_{Ckl \in N_r(ij)} A(ij;kl) X_{kl}(t) Y_{kl}(t) \quad (2.1)$$

The feedback template $A(ij;kl)$ is a function of both the state and output voltages of the neighbourhood cells $N_{i,j}$, including the cell itself. The feedback template for a 2-neighbourhood CNN is given by the matrix

$$\begin{bmatrix} A_{-2,-2} & A_{-2,-1} & A_{-2,0} & A_{-2,1} & A_{-2,2} \\ A_{-1,-2} & A_{-1,-1} & A_{-1,0} & A_{-1,1} & A_{-1,2} \\ A_{0,-2} & A_{0,-1} & A_{0,0} & A_{0,1} & A_{0,2} \\ A_{1,-2} & A_{1,-1} & A_{1,0} & A_{1,1} & A_{1,2} \\ A_{2,-2} & A_{2,-1} & A_{2,0} & A_{2,1} & A_{2,2} \end{bmatrix} \quad (2.2)$$

2.2.1 Binary Addition

Consider the addition of two N-bit binary numbers $A = a_1 a_2 \dots a_N$ and $B = b_1 b_2 \dots b_N$ [20]. For this we need a $3 \times N$ CNN. The initial values of the state voltages of the cells in the first row are set to -1 and the values of A and B are loaded as the initial values of the state voltages of the second and third row respectively. After the

transient time the result of the addition is stored in the third row. The binary addition is based on the following two rules:

2.2.2 Rule 1:

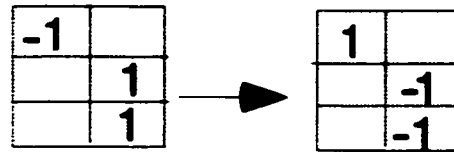


Figure 2.2 Application of rule 1

Figure 2.2 shows the application of binary addition Rule 1. It shows that the output voltages of any two adjacent cells in a column at position $(2,j)$ and $(3,j)$ [being referred to as $Y_u(t)$ and $Y_d(t)$ respectively] with initial voltages of 1 will drop to -1. Simultaneously, the output voltage of the cell at position $(1,j-1)$ (being referred to as $Y_c(t)$) will rise from -1 to 1. The ideal time for applying Rule 1 is indicated when $V_{c1} = Y_d(t) - Y_u(t)$ is zero.

2.2.3 Rule 2:

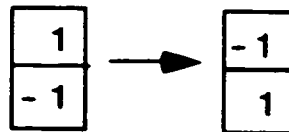


Figure 2.3 Application of rule 2

If there are any two adjacent cells in a column with the output voltage of the upper cell $Y_u(t)$ equal to 1 and the output voltage of the lower cell $Y_d(t)$ equal to -1, then the value

of the upper cell will drop to -1 and, simultaneously, the value of the lower cell will rise to 1 as shown in Figure 2.3. The ideal time for applying Rule 2 is indicated when $V_{c2} = |Y_u(t) + Y_d(t)|$ is zero.

These rules are applied successively without the need to change the templates, using an external controller; hence they are called as self-programming templates. The addition is finished as soon as all the output voltages of the cells in the first two rows settle down to -1. The final result is stored as the output voltages of the cells in the last row.

2.3 Self- Feedback Template

We use a $3 \times N$ CNN with self-programming feedback template given by the Eqn. (2.3)

$$Af = \begin{bmatrix} 0 & A_{-2,-1} & 0 & 0 & 0 \\ 0 & A_{-1,-1} & A_{-1,0} & 0 & 0 \\ 0 & 0 & A_{0,0} & 0 & 0 \\ 0 & 0 & A_{1,0} & A_{1,1} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.3)$$

From this, we can group the template elements that enforce the two rules as:

Group 1: $[A_{-2,-1}, A_{-1,-1}, A_{1,1}]$ enforcing rule 1.

Group 2: $[A_{-1,0}, A_{1,0}]$ enforcing rule 2.

$A_{0,0}$ is the self feedback element which is a linear function of the output voltage of the cell and is given by the Eqn. (2.4).

$$A_{0,0} = k \times Y_{ij}(t) \quad (2.4)$$

At any point in time, any given cell is subject to the application of only one rule and consequently only one element of its feedback template, in addition to the self-feedback element, is non-zero.

The non-zero elements which are used to apply rule 1 and rule 2 can be written as the product of two terms as:

$$A_{m,n} = F_{m,n} \times H_{m,n} \quad (2.5)$$

The $H_{m,n}$ term represents a hysteresis characteristic shown in Figure 2.4 which is controlled by V_{c1} for Group 1 or V_{c2} for Group 2.

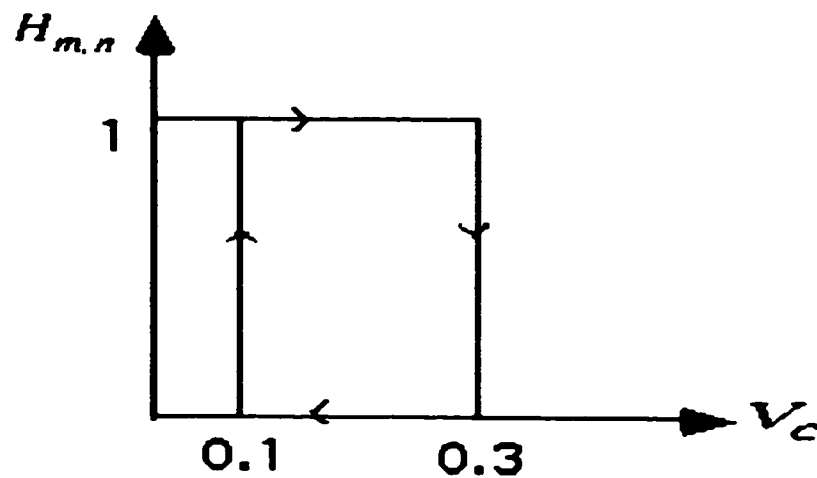


Figure 2.4 Hysteresis characteristic of $H_{m,n}$

The $F_{m,n}$ and their corresponding V_{c1} for elements of Group 1 are given in Eqn. (2.6), Eqn. (2.7) and Eqn. (2.8).

(2.6)

$$F_{-2,-1} = \begin{cases} -k*(1+x_{ij}) & \text{if } (y_{i-2,j-1} + y_{i-1,j} + y_{i,j} \geq -0.5) \\ 0 & \text{otherwise} \end{cases}$$

$$V_{c1} = Y_{i,j} - Y_{i-1,j}$$

$$F_{-1,-1} = \begin{cases} -k \times (1 + X_{ij}) & \text{if } Y_{i-2,j-1} + Y_{i-1,j} + Y_{i,j} \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (2.7)$$

$$V_{c1} = Y_{i+1,j} - Y_{i,j}$$

$$F_{1,1} = \begin{cases} k*(1-x_{ij}) & \text{if } (y_{i,j} + y_{i+1,j+1} + y_{i+2,j+1} \geq -0.5) \\ 0 & \text{otherwise} \end{cases} \quad (2.8)$$

$$V_{c1} = Y_{i+2,j+1} - Y_{i+1,j+1}$$

The $F_{m,n}$ and their corresponding v_{c2} for elements of Group 2 are given in Eqn. (2.9) and Eqn. (2.10).

$$F_{-1,0} = \begin{cases} k*(1-x_{ij}) & \text{if } (y_{i,j} \leq 0.5) \\ & \& (|y_{i+1,j}| \geq 0.8) \\ & \& (y_{i-2,j-1} \leq -0.5) \\ 0 & \text{otherwise} \end{cases}$$

$$v_{c2} = |Y_{ij} + Y_{i-1,j}| \quad (2.9)$$

In the above equation, the second and third conditions are applied only to the cells of the second and third rows respectively.

$$F_{-1,0} = \begin{cases} k*(1-x_{ij}) & \text{if } (y_{i,j} \leq 0.5) \\ & \& (y_{i+1,j} \geq 0.8) \\ & \& (y_{i-2,j-1} \leq -0.5) \\ 0 & \text{otherwise} \end{cases}$$

(2.10)

$$V_{c2} = |Y_{ij} + Y_{i+1,j}|$$

In the above equation, second condition is applied only to the cells of the first row

2.4 Equivalent Circuit Diagram of a Cell

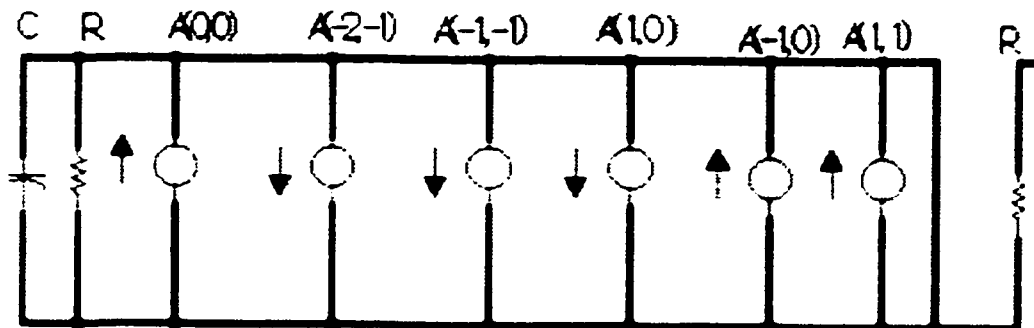


Figure 2.5 Equivalent circuit diagram

Figure 2.5 shows the equivalent circuit diagram of the cell. The difference between this and the basic cell diagram in Figure 1.4 is that, in this application, we do not need the feedforward templates. Also we note that among the feedback templates some of them are

used to charge the capacitor (current sourcing) while the others discharge the capacitor (current sinking).

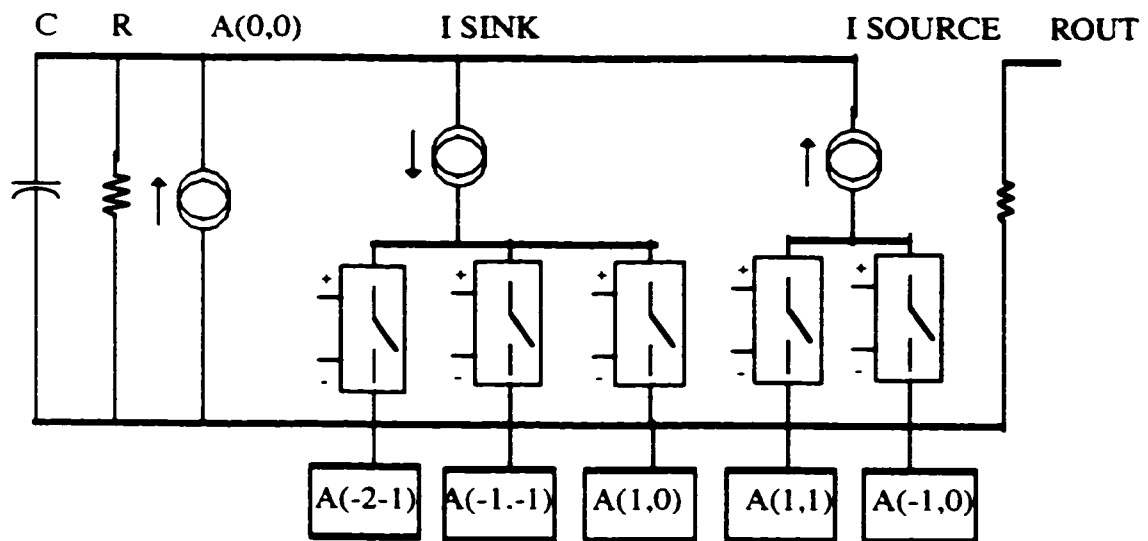


Figure 2.6 Detailed equivalent diagram of a cell

Chapter 3

VLSI IMPLEMENTATION

The VLSI implementation is accomplished by realizing the concepts described in the previous chapter. From Figure 2.1, it is possible to draw the building blocks for image processing applications. As reported earlier[4], the basic CNN cell consists of the cell resistor, cell capacitor, nonlinear voltage controlled current source and an output resistor. These blocks can be realized by choosing the appropriate circuits described in the following section. This CNN structure also provides programming capability using current scaling circuits. The VLSI building blocks for the binary arithmetic application are described in a later section.

3.1 Nonlinear V/I Converter

The non-linear converter is basically an operational transconductance amplifier (OTA)[21][22][23]. Figure 3.1 shows the circuit diagram of a simple OTA. It generates an output current I_{out} that is a function of the differential input voltage ($V_1 - V_2$). It essentially consists of a differential pair and a current mirror. The bottom transistor is used as a current source; its drain voltage, V , is large enough that the drain current, I_b , is saturated at a value set by the gate voltage, V_b .

Referring to Figure 3.1, currents flowing through transistors M1 and M2 are given by:

$$I_1 = \frac{\beta}{2}(V_{gs1} - V_T)^2 \quad (3.1)$$

and

$$I_2 = \frac{\beta}{2}(V_{gs2} - V_T)^2 \quad (3.2)$$

Therefore, the bias current is: $I_b = I_1 + I_2$

Notice that if V_1 is more positive than V_2 , transistor M2 is turned off, thus all of the current flows through M1 and hence I_1 is approximately equal to I_b , and I_2 is approximately equal to zero. Conversely, if V_2 is more positive than V_1 M1 is turned off, I_2 is approximately equal to I_b and I_1 approximately equal to zero[24][25][26].

The output current of the simple transconductance amplifier ($I_{out} = I_1 - I_2$) is:

$$I_{out} = \sqrt{I_b \beta} (V_1 - V_2) \quad (3.3)$$

The output current, plotted as a function of $(V_1 - V_2)$, is shown in Figure 3.2. The transconductance, G_m , of the amplifier is the slope of the curve at the origin and it can be shown to be[27][28]:

$$G_m = \frac{\partial I_{out}}{\partial V_{in}} = \sqrt{2\beta I_b \frac{W}{L}} \quad (3.4)$$

where β is the transconductance parameter of the MOS transistor[29][30].

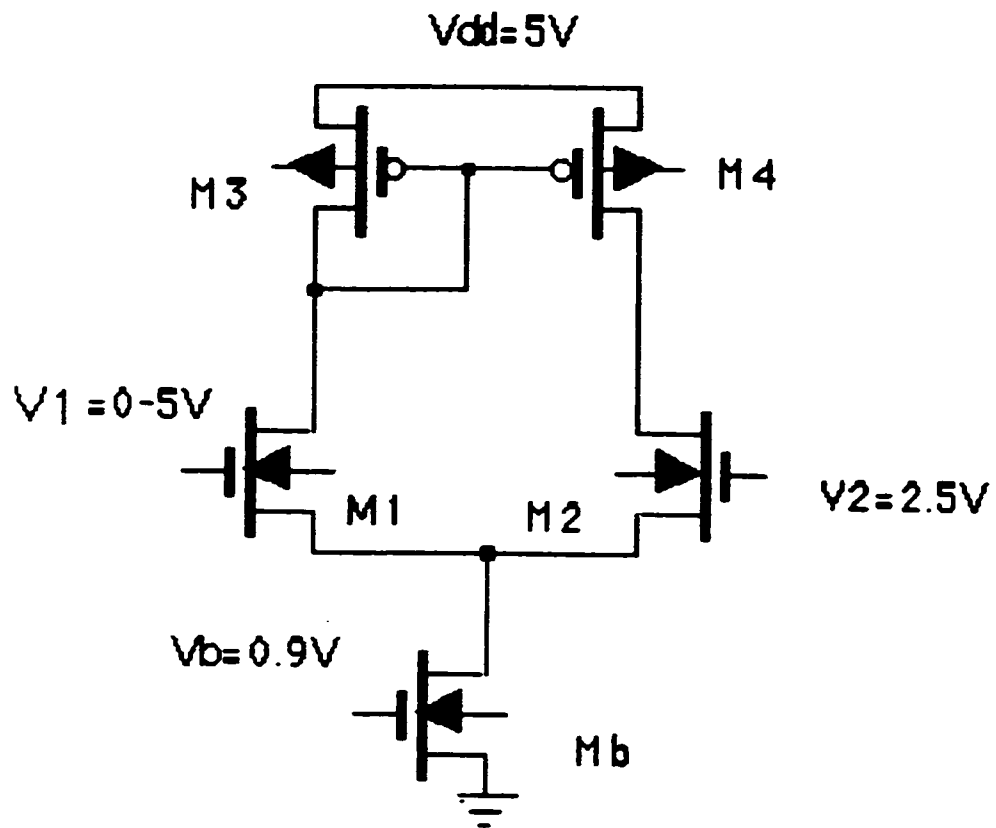


Figure 3.1 Simple transconductance amplifier

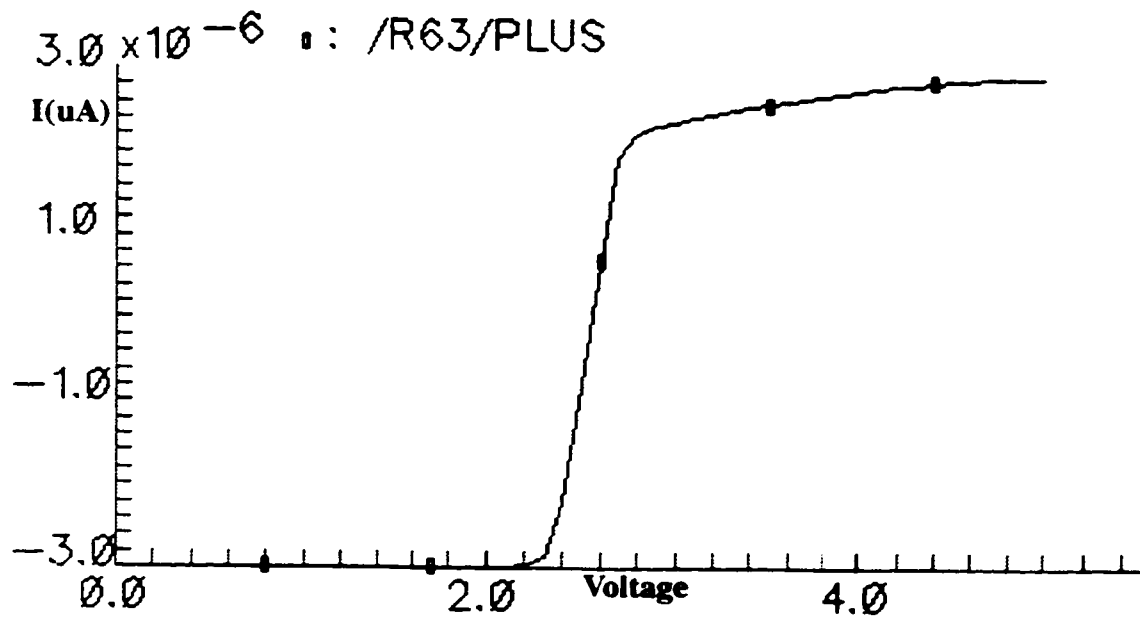


Figure 3.2 DC response of OTA

3.1.1 Programmable Current Scaling Circuit

A very compact tunable current scaling circuit is shown in Figure 3.3[10]. Using this circuit the value of the A template elements can be varied within the prescribed dynamic range (DR)[11]. The dynamic range of a variable is defined as its largest possible value divided by its smallest possible value.

The transistors M5 and M6 operate in the non-saturation region as resistors, and transistors M3 and M4 operate in saturation as voltage followers. Notice that at both the input and output, current mirrors are used to reflect the input and the output currents respectively. When the gate voltage of M5 is equal to that of M6 i.e., zero voltage, then the source voltages of M3 and M4 will become equal, hence $I_{in} = I_{out}$. When the gate voltage of M6 is greater than M5, then the source voltage of M4 increases and its gate-source voltage decreases. So I_{out} is scaled down to a value depending on the gate voltage of M6.

The current scaling factor, F , can be calculated by considering the input-output relationship

$$F = \frac{I_{out}}{I_{in}} = \frac{1/g_{ds5} + 1/g_{m3}}{1/g_{ds6} + 1/g_{m4}} \quad (3.5)$$

If g_m of M3 and M4 is much larger than g_{ds} of M5 and M6, then the scaling factor is proportional to the ratio of the resistances.

Therefore the template value can be calculated by multiplying the scaling factor with the transconductance of the differential pair.

Cascading of the nonlinear V/I converter with the scaling circuit is shown in Figure 3.4. The positive current flowing out of the OTA is scaled down by the circuit shown in Figure 3.3. The negative current is scaled by the complement of the positive scaling

circuit. The output response of the circuit is shown in Figure 3.5. Notice that the maximum current flow is when the voltage (V_{value}) is minimum.

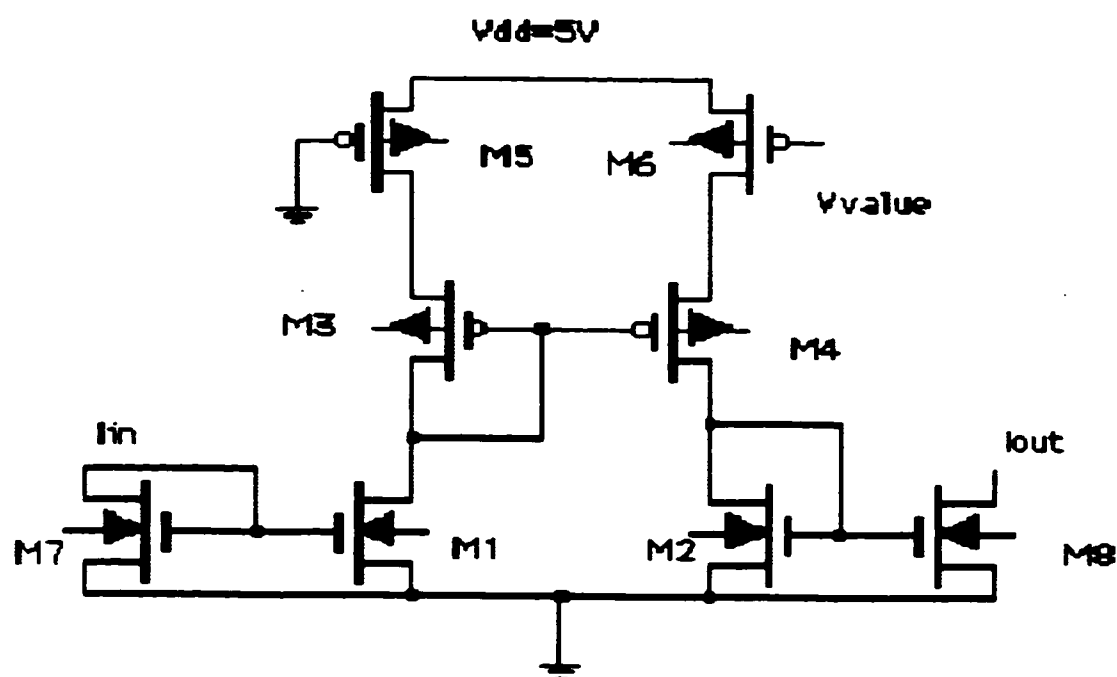


Figure 3.3 Current scaling circuit for nonlinear V/I converter

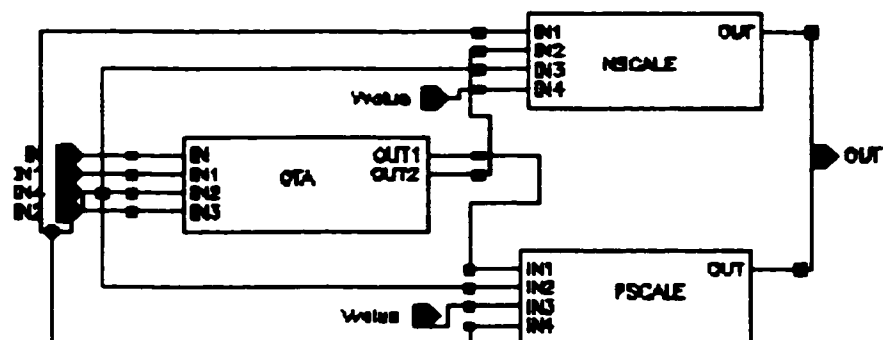


Figure 3.4 Cascading of nonlinear V/I converter with a current scaling circuit

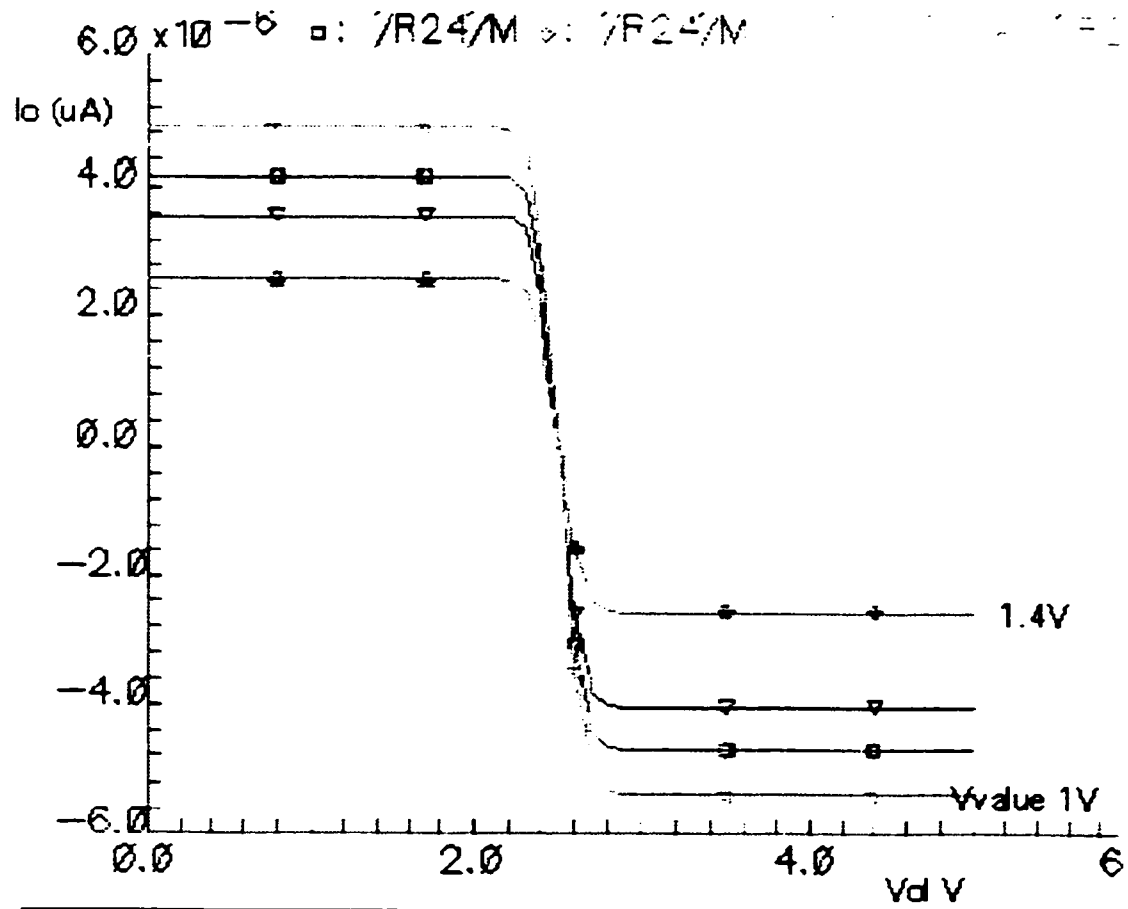


Figure 3.5 Output response of the scaling circuit for different *Vvalue* Voltages

3.2 Tunable Linear V/I Converter

A tunable linear V/I converter is used to control the B template value of the CNN. A linear V/I converter is designed with the use of a linear composite n-channel transistor and CMOS composite transistors[31][32]. Figure 3.6 shows n-channel linear transistors. It consists of two matched transistors, M1 and M3, and a load transistor, M2 (which is matched with M1 and M3). With the threshold voltage, V_T , and the transconductance parameter, β , the currents I_{d1} and I_{d2} can be expressed as

$$I_{d1} = \frac{\beta}{2}(V_{in} - V_x - V_T)^2 \quad (3.6)$$

and

$$I_{d2} = \frac{\beta}{2}(V_x - V_c - V_T)^2 \quad (3.7)$$

where all the transistors operate in the saturation region. Because M1 and M3 are matched, $V_{in} - V_x = V_x - V_{ss}$. Thus:

$$2V_x = V_{in} + V_{ss} \quad (3.8)$$

The difference in the currents is then given by:

$$I_{d1} - I_{d2} = \frac{\beta}{2}(V_c - V_{ss})(V_{in} - V_c - 2V_T) \quad (3.9)$$

Eqn. (3.9) shows that the differential current and the input voltage have a linear relationship with a DC voltage offset. Therefore, a linear transconductance can be implemented by cross-coupling two basic cells as shown in Figure 3.7. The differential current is then given by:

$$I_1 - I_2 = \frac{\beta}{2}(V_c - V_{ss})(V_1 - V_c - 2V_T) - \frac{\beta}{2}(V_c - V_{ss})(V_2 - V_c - 2V_T) \quad (3.10)$$

$$= \frac{\beta}{2}(V_c - V_{ss})(V_1 - V_2) \quad (3.11)$$

This new linear transconductance is a constant G_m and is given by:

$$G_m = \frac{\beta}{2}(V_c - V_{ss}) \quad (3.12)$$

Although it can be tuned (programmed) by a voltage V_c , a disadvantage of this circuit is that the linear input range is limited to $V_{ss} + 2V_T \leq V_{1,2} \leq V_{DD} + V_T$, where $V_{ss} + 2V_T$ is the voltage required to turn on transistors M0, M5, M3 and M8.

To increase the linear range, the circuit is modified by replacing M1 and M2 with CMOS composite transistors as shown in Figure 3.8, where the equivalent transconductance parameter and the threshold voltage are given by Eqn. (3.13)[33]:

$$\left[\frac{1}{\sqrt{\beta_{equ}}} \right] = \left[\frac{1}{\sqrt{\beta_n}} \right] + \left[\frac{1}{\sqrt{\beta_p}} \right] \quad (3.13)$$

and

$$V_{Teq} = V_{Tn} - V_{Tp} \quad (3.14)$$

Since the transconductance parameter β_p is much larger than β_n , $\beta_{equ} = \beta_n = \beta$ (say), then:

$$I_1 - I_2 = \frac{\beta}{2}(V_c - V_{ss} - V_T - V_{Teq})(V_1 - V_2) \quad (3.15)$$

and

$$G = \frac{\beta}{2}(V_c - V_{ss} - V_T - V_{Teq}) \quad (3.16)$$

The output response of the circuit i is shown in Figure 3.9:

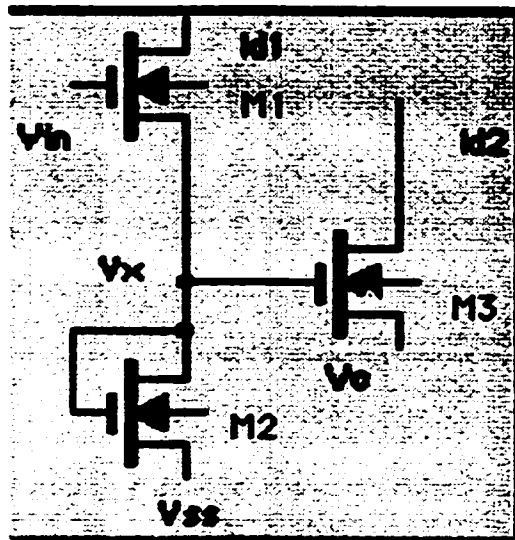


Figure 3.6 Basic cell of linear transconductance circuit

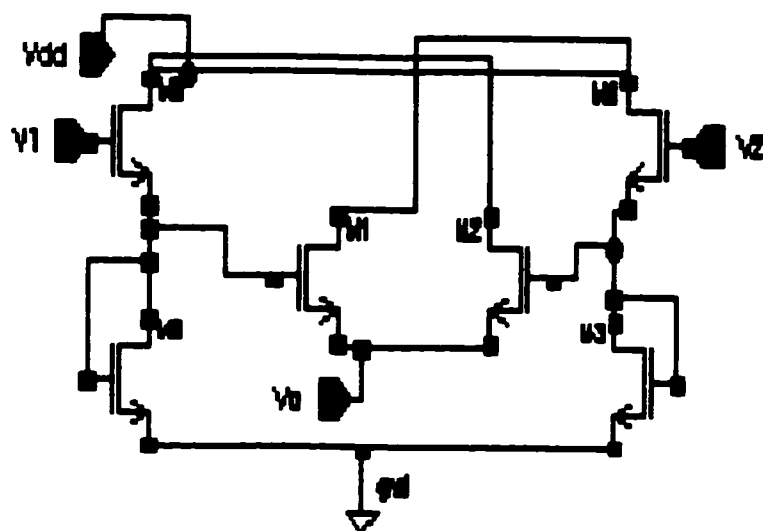


Figure 3.7 Linear tunable transconductance circuit

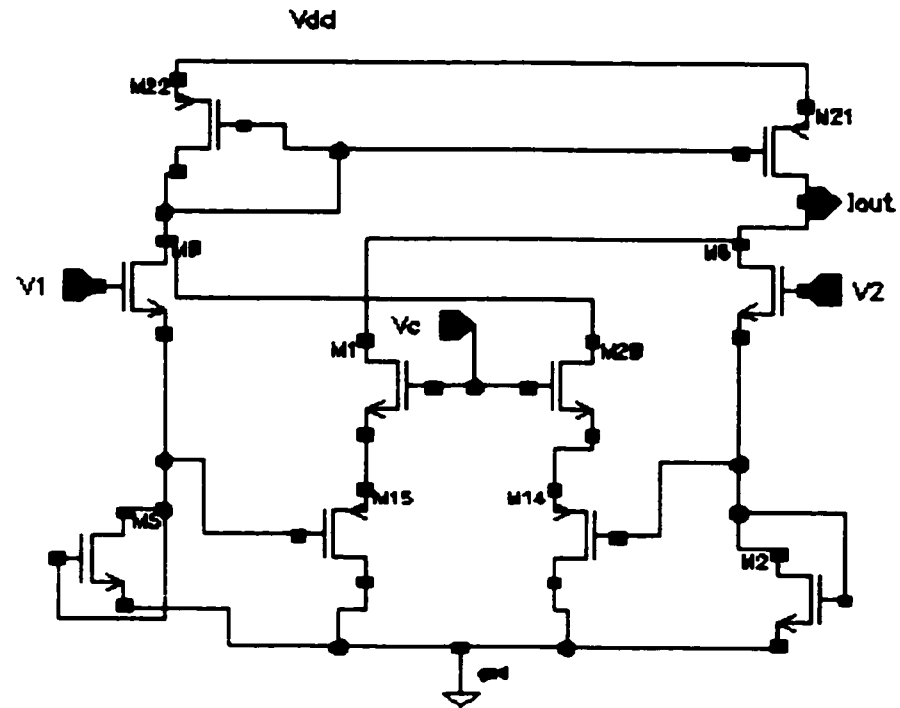


Figure 3.8 Modified linear tunable transconductance circuit

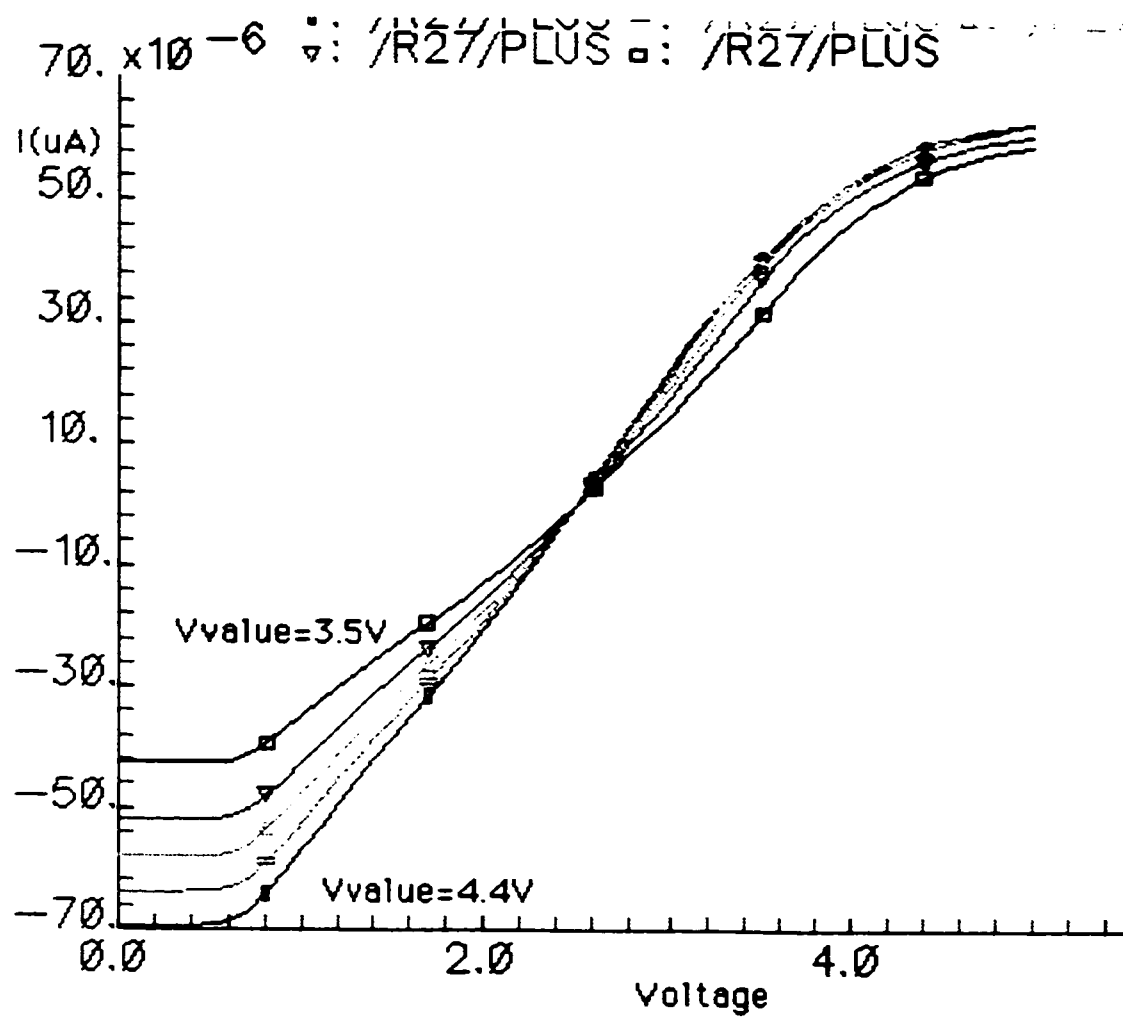


Figure 3.9 Output response of a linear transconductance circuit

3.3 Cell Resistor and State Capacitor

The cell time constant is defined by a cell resistor R and a state capacitor C . The cell resistor is built with an NMOS transistor and a PMOS transistor connected in parallel and biased in the linear region as shown in Figure 3.10[34]. This compact structure is a grounded resistor for all state voltages and has a constant resistance in the unit range. In this structure, input $V3$ is varied from 0-5 volts. When $V3$ is less than the bias voltage($V2$), the source and drain of both the NMOS and PMOS transistor are interchanged and hence negative currents flow through both NMOS and PMOS transistors. When $V3$ is greater than $V2$, current flows through the transistors in a reverse manner. The plot of this current with $V3$ gives the resistance plot shown in Figure 3.11.

3.3.1 State capacitor

For a parallel plate capacitor assumption, the capacitance is represented as:

$$C = \epsilon \frac{A}{d} \quad (3.17)$$

where A is the area of the electric plate, d is the distance between two plates and ϵ is the dielectric constant.

A capacitor can be designed by several different techniques using standard CMOS technology. One method uses double polysilicon plates; however, this approach requires large silicon area due to large oxide thickness between the polysilicon and hence low sheet capacitance. A second method uses the gate oxide capacitance of the MOS transistor in the ohmic region. This approach requires less silicon area than the first method because of the thin oxide layer of the gate. This technique is adopted in our design[35].

In order to use a MOS 'sandwich' as a capacitor, the channel has to be biased in the strong inversion region. This is done by ensuring a minimum gate-source voltage of several hundred millivolts above the threshold voltage, V_T .

Gate oxide capacitance C_{gox} can be represented as

$$C_{gox} = AC_{ox} = A \frac{\epsilon_{ox}}{T_{ox}} \quad (3.18)$$

where

A $= (W * L)$ is area of the transistor

C_{ox} $=$ capacitance density of the channel junction

ϵ_{ox} $=$ dielectric constant of the gate oxide

T_{ox} $=$ Thickness of the gate oxide

Therefore the transistor area is given by:

$$A = \frac{C}{C_{ox}} = C \frac{T_{ox}}{\epsilon_{ox}} \quad (3.19)$$

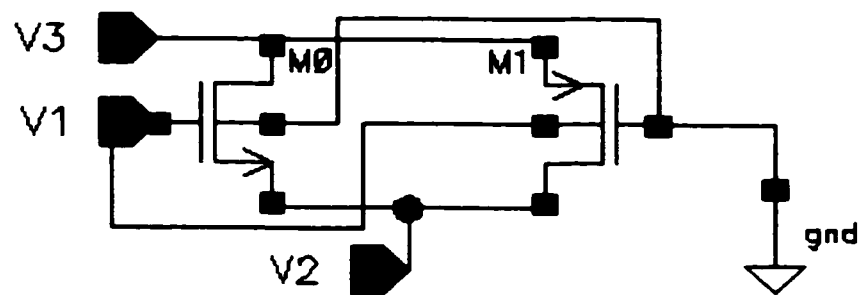


Figure 3.10 Cell resistor circuit

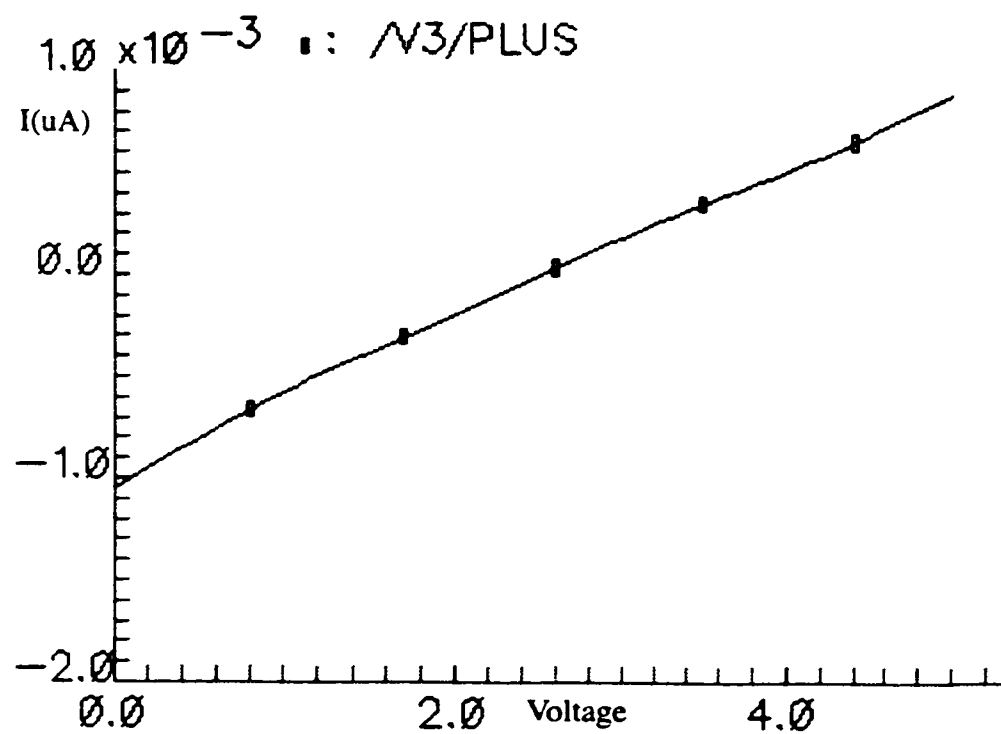


Figure 3.11 DC. response of the cell resistor

3.4 VLSI Library Cells for Binary Addition

Cellular neural networks (CNNs) can be used to perform arithmetic operations. The algorithms used to implement these concepts are introduced in chapter 2[20]. From the analysis of the algorithms, we required to have appropriate building blocks to implement this idea in VLSI. The following sections briefly describes the circuits and their layouts of the required building blocks. These layouts are verified by conducting DRC, LVS and post layout simulations. Finally the basic CNN cell circuit and a demonstration of the verification of Rule 2 for the binary arithmetic application is presented.

3.4.1 Absolute Value Function Circuit

In many system applications, it is not the signed value of the signal that is important, but rather the absolute value of the signal. If the signal is current, it is extremely easy to take the absolute value with a current mirror. A p-channel current mirror can act as a current source, whereas an n-channel current mirror can act as a current sink. The positive or negative part of any signal represented as a bidirectional current can thus be taken by the appropriate current mirror. There are circuits which create a current proportional to the absolute value of the voltage difference[21]. In this application the absolute value of a current signal can be taken by placing appropriate current mirrors as shown in Figure 3.12. Here the current mirror formed by M6 and M7 is used to reflect the positive current and the current mirror formed by M1 and M2 is used to reflect the negative current.

The output current is:

$$I_{out} = |I_1 + I_2| \quad (3.20)$$

Figure 3.13 and Figure 3.14 show the DC response and the layout, respectively, of the absolute value function circuit.

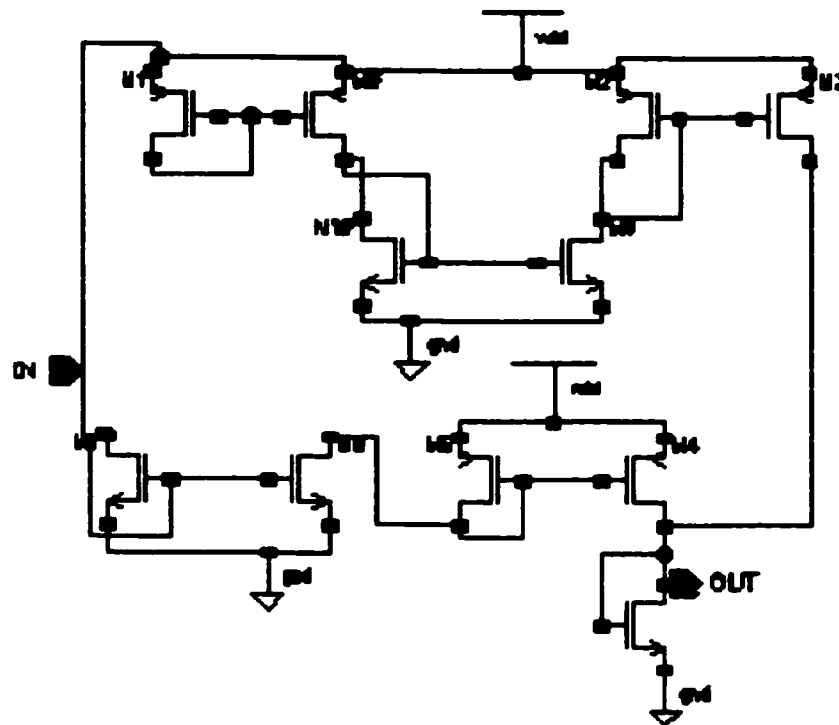


Figure 3.12 Absolute value function circuit

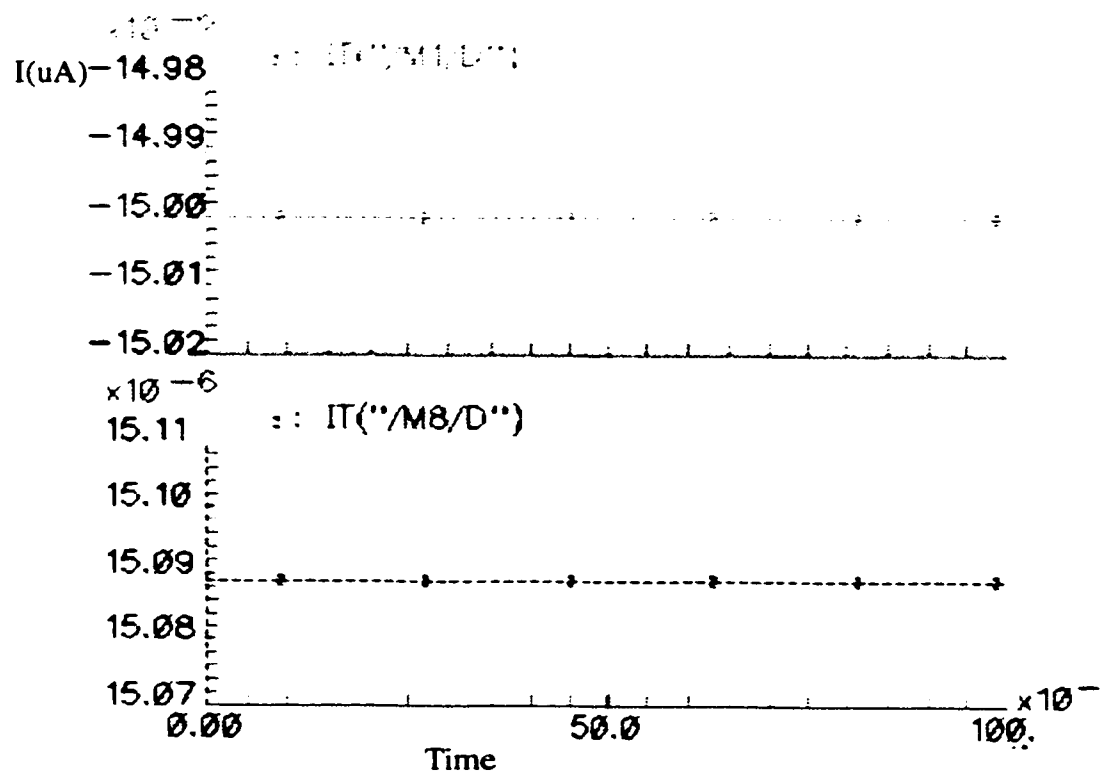


Figure 3.13 DC response of the absolute value circuit

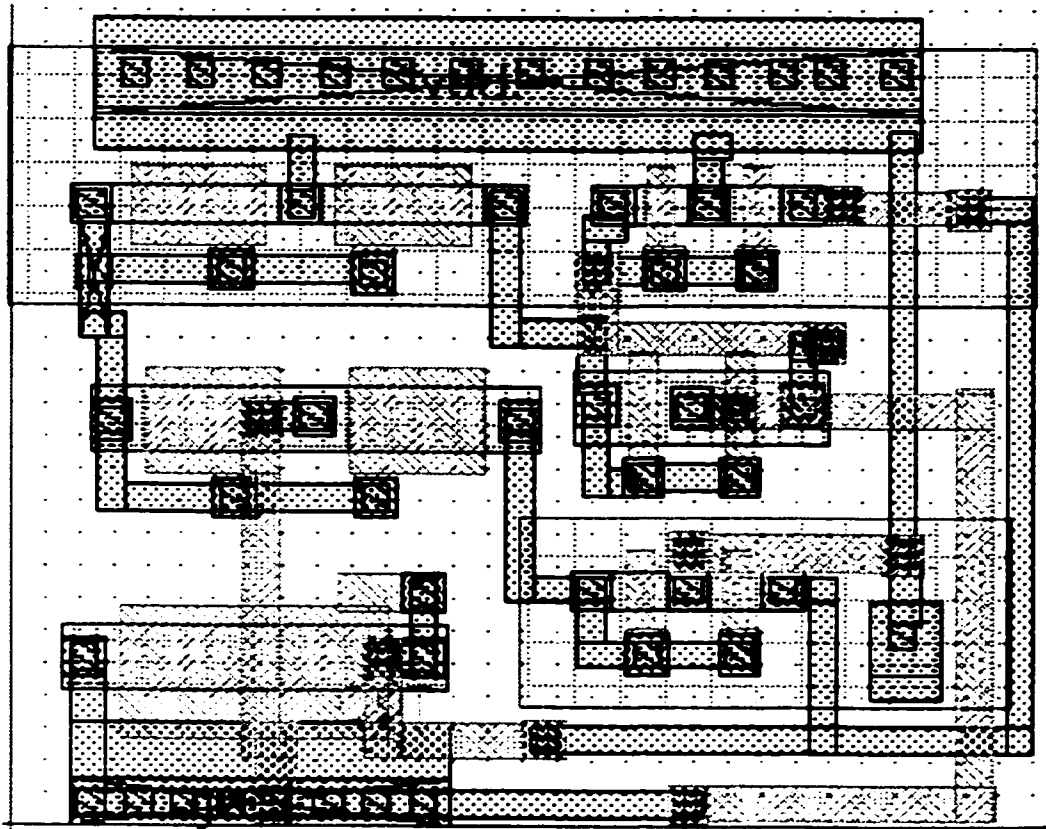


Figure 3.14 Layout of the absolute value circuit

3.4.2 Current Comparator

The circuit diagram for the current comparator is shown in Figure 3.15. It is basically a modified version of the maximum-minimum circuit [36]. It consists of two cross coupled current mirrors which reflect two input currents to be compared. The cross coupled connection ensures that there is a drain voltage reduction or increase at the receiving transistors of the two current mirrors, as shown in Figure 3.15. This will cause an increase or decrease of the gate voltage in the current mirrors. If we connect inverters with appropriate sizes to these current mirrors, we obtain an output voltage with two logic levels, hence providing the comparison. Figure 3.16 shows the DC response and Figure 3.17 shows the layout of the current comparator[37][38].

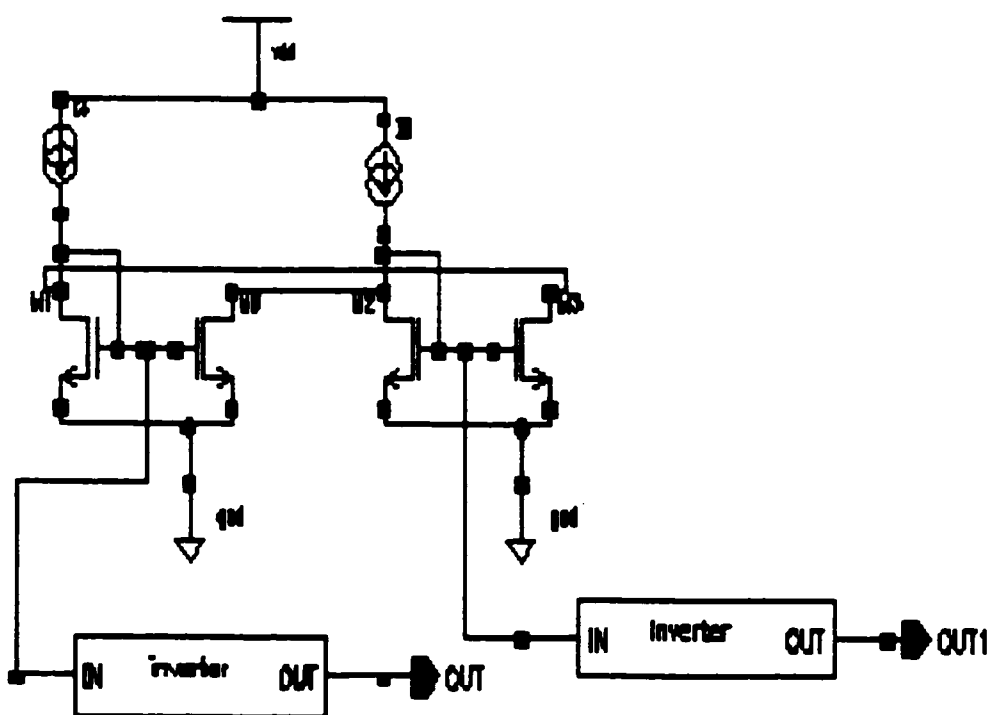


Figure 3.15 Current comparator

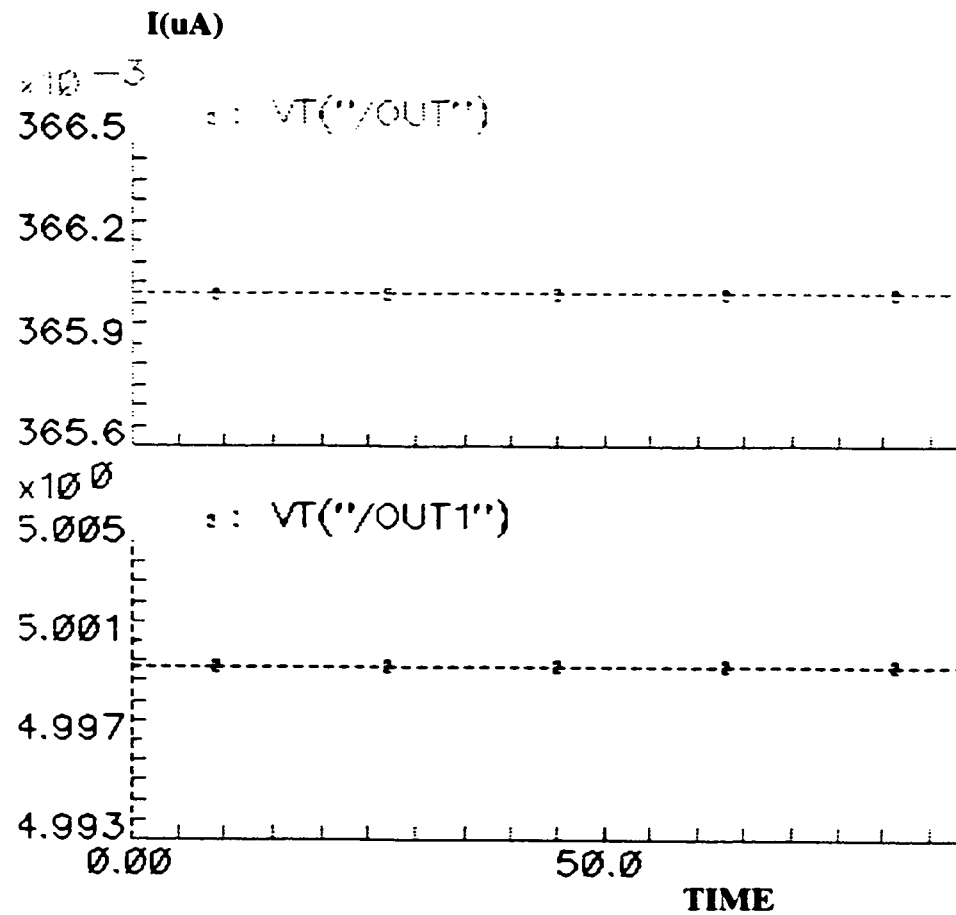


Figure 3.16 DC response of the current comparator

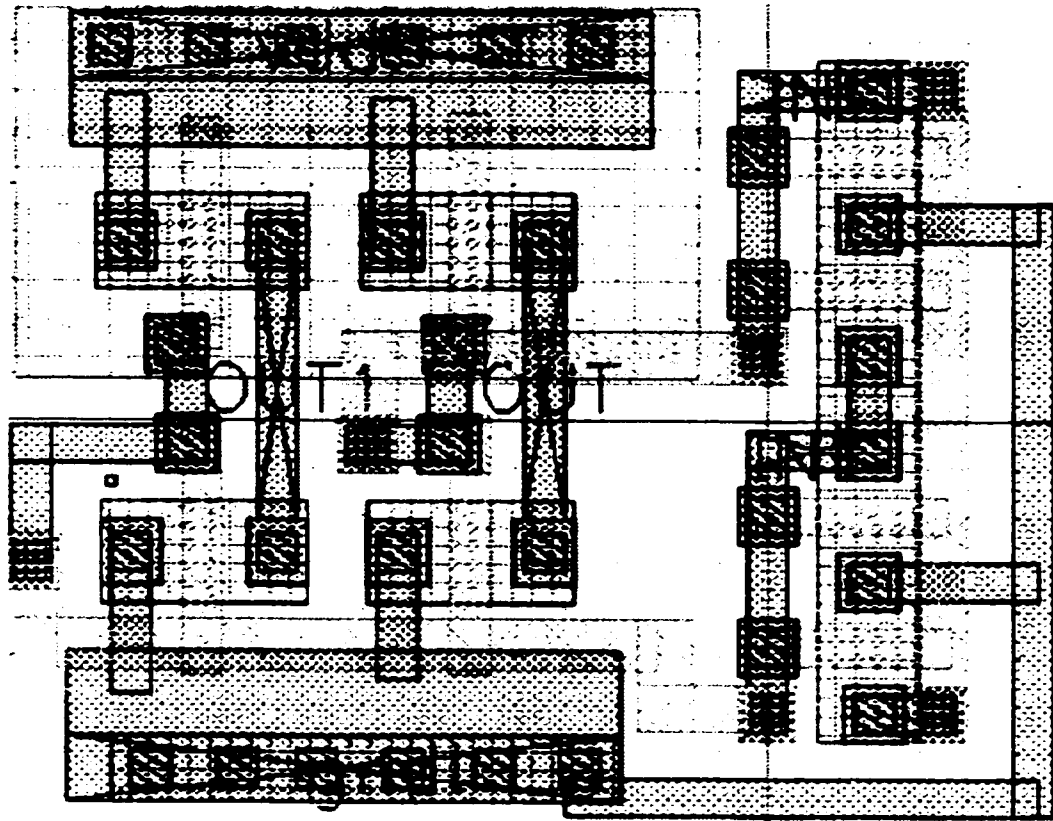


Figure 3.17 Layout of current comparator

3.4.3 Hysteresis Circuit

In the hysteresis circuit, shown Figure 3.18, the bottom circuit M4, M5, and M0 (which is called the N-subcircuit), is loaded by the top circuit, M1, M2, and M3 (the P-subcircuit). As a result of the circuit symmetry, each circuit acts as a nonlinear load to the other.

In order to understand the circuit operation, assume that the voltage V_g in Figure 3.18 is zero. Then transistors M4 and M0 are off. Transistors M1 and M3 are in the linear mode of operation, but the voltage drop at each is zero, because the current in M1 and M2 is equal to the current in M4 and M0. The output voltage is equal to V_{DD} (or high). Transistor M5 is on since its drain and gate have the same voltage (V_{DD}). When V_g rises above V_{TN} , transistor M4 turns on and starts to conduct. The current in M4 is determined by Eqn. (3.21):

$$I_{D4} = \beta_4(V_g - V_{TN})^2 \quad (3.21)$$

If the potential, V_N , decreases, the triggering operation starts when the voltage V_g reaches V_{Hi} . At this point, due to the simultaneous increase of V_g and decrease of V_N , transistor M0 turns on. The detailed description of the circuit can be found in the references[39][40][41]. It is possible to obtain a relationship between the transistor parameters and the triggering operation, i.e. V_{Hi} and V_{Li} can be specified by the following equations:

$$\frac{\beta_4}{\beta_5} = \left(\frac{V_{DD} - V_{Hi}}{V_{Hi} - V_{TN}} \right)^2 \quad (3.22)$$

$$\frac{\beta_1}{\beta_3} = \left(\frac{V_{Li}}{V_{DD} - V_{Li} - |V_{TP}|} \right)^2 \quad (3.23)$$

where β_i is the transconductance parameter of the transistor ($\beta = \mu C_{ox} \frac{W}{L}$). V_{TN} and V_{TP} are the threshold voltages of NMOS and PMOS transistors respectively.

Eqn. (3.22) and Eqn. (3.23) give sufficient information to design the hysteresis circuit. Figure 3.19 shows the D.C. response of the circuit when sweeping the input voltage from 0-5 V and 5-0 V. Figure 3.20 shows the transient response of the circuit displaying the triggering points (V_{Hi} and V_{Li}), when applied by a triangular pulse and observe the triggering points. Figure 3.21 shows the layout of the hysteresis circuit.

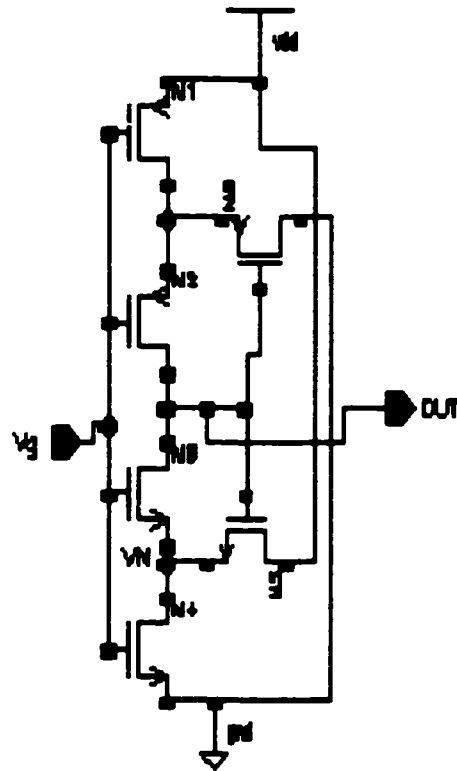


Figure 3.18 Hysteresis function circuit

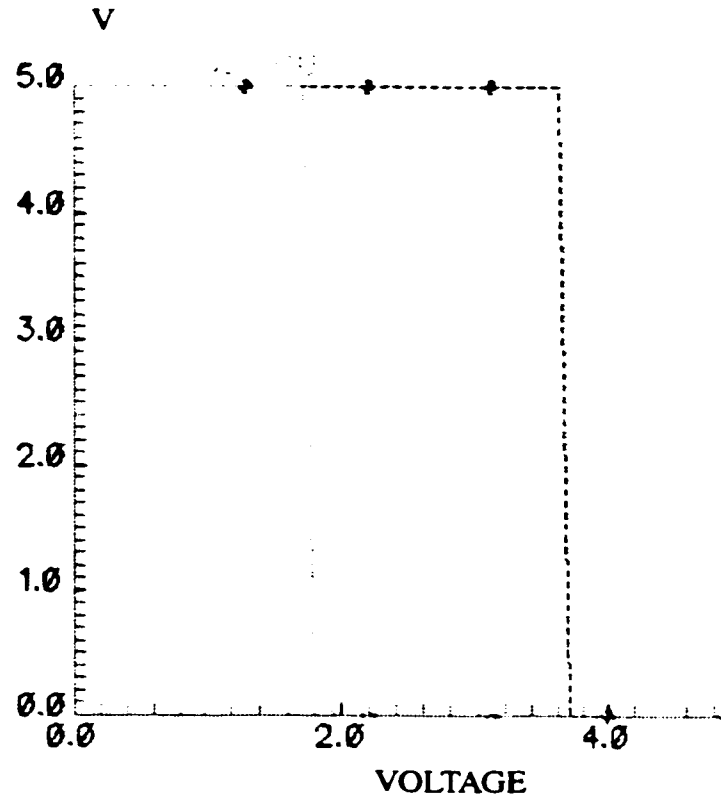


Figure 3.19 DC. response of the hysteresis circuit

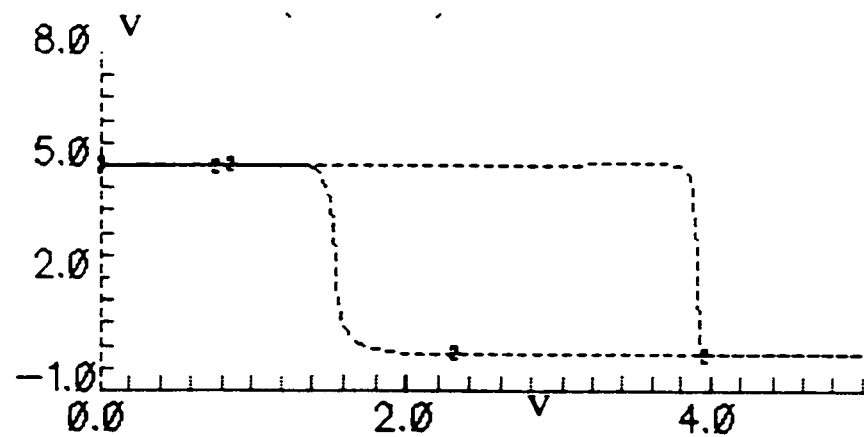


Figure 3.20 Transient response of the circuit

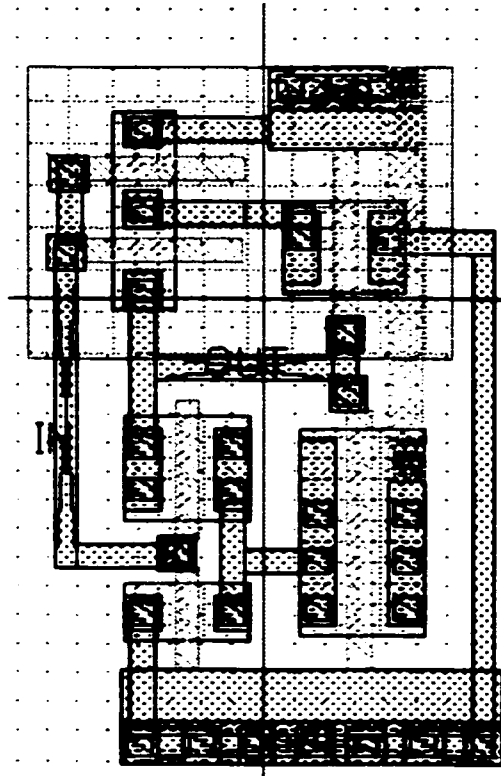


Figure 3.21 Layout of the hysteresis circuit

3.4.4 Current Adder and Subtractor Circuits

Addition and subtraction on current type signals are obtained by following Kirchhoff's current law[21]. This law states that the algebraic sum of the currents into an electrical node is zero, i.e. the sum of the currents out of the node is same as the sum of the currents into the node. By using this basic principle, it is possible to construct current adders and subtractors. The current addition is carried out by using two current mirrors and a load. The two current mirrors are used to reflect the two input currents to be added; the circuit for current addition is shown in Figure 3.22. The output can be measured at the load ($I_{out} = I_1 + I_2$); the simulation result is shown in Figure 3.25. The current subtraction can be carried out in a similar fashion, and the circuit is shown in Figure 3.23. The subtracted value can be measured at the output node ($I_{out} = I_1 - I_2$), the result is shown in Figure 3.24.

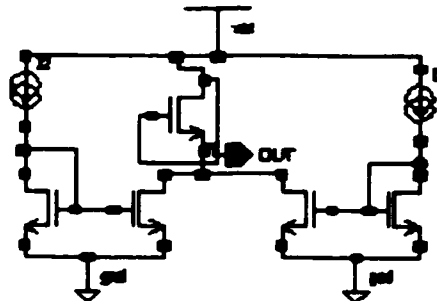


Figure 3.22 Current adder

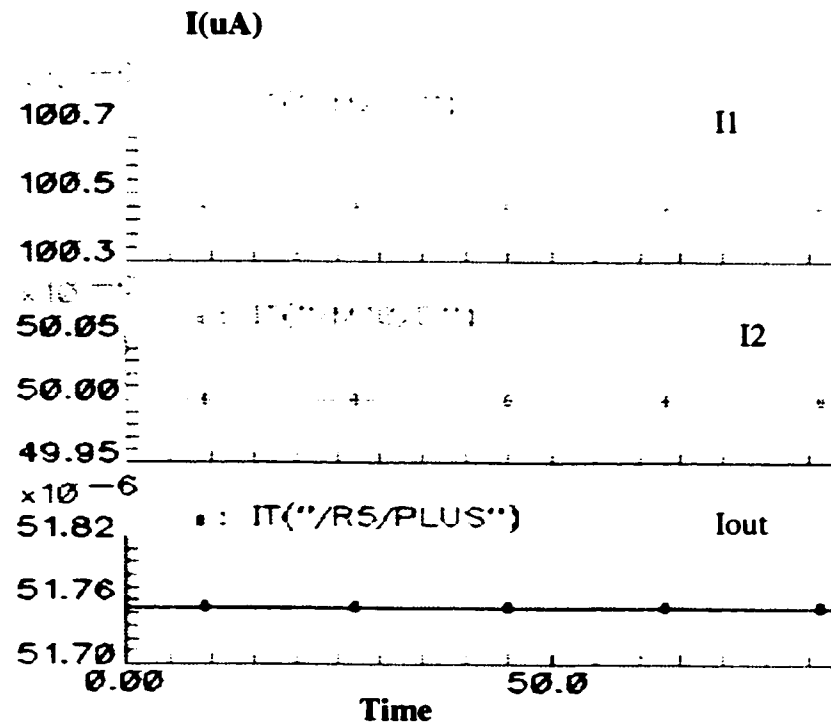


Figure 3.24 DC response of current subtractor

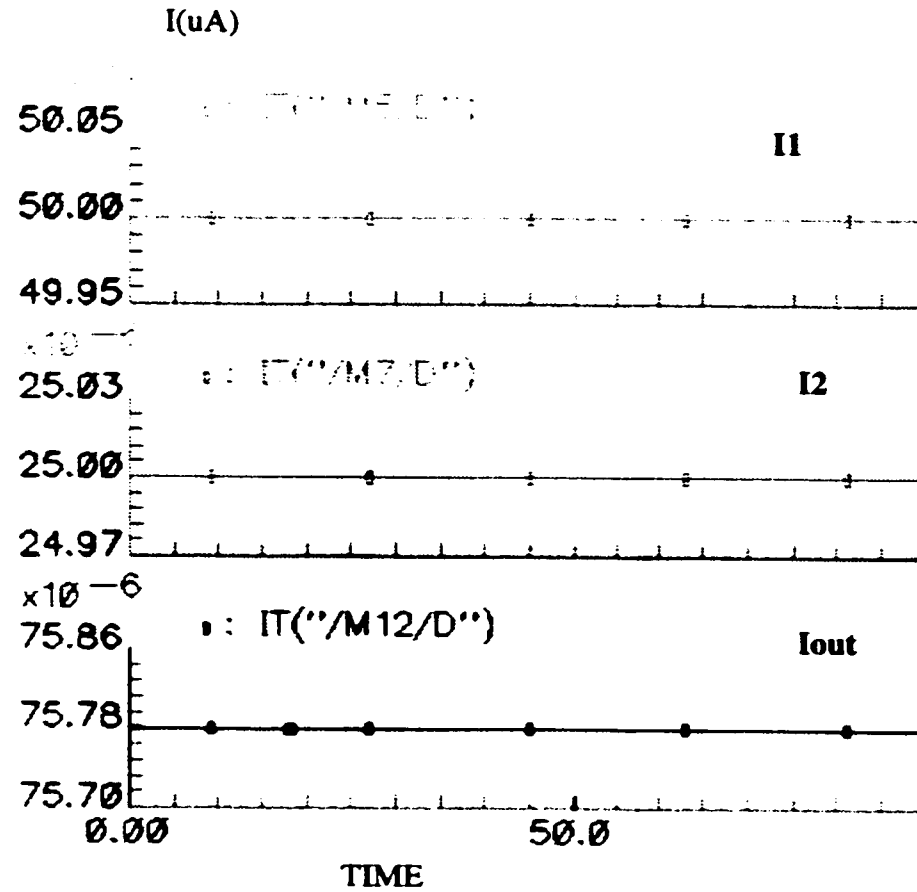


Figure 3.25 DC response of current adder

3.5 Basic CNN Cell

Figure 3.26 shows the basic CNN cell. As explained in the previous chapters, a basic CNN cell consists of a cell resistor, cell capacitor, and voltage controlled current source. This structure is designed for arithmetic operations. The operation of the cell is described below.

Logic 1 and logic 0 are determined by initial capacitor voltages of 4V and 0.5V respectively. Suppose that 4V is assigned to the state capacitor and the state voltage is connected to one of the inputs of the OTA. If $V_1 > V_2$, more current will flow through transistor M5 and less current will flow through transistor M55. As can be seen, the feedback is connected to the state voltage node, a decrease in current flow through M55 causes the feedback voltage to increase up to $V_{DD} - V_T$, so the state voltage will rise to $V_{DD} - V_T$. This will cause the output current to flow in the positive direction which indicates the logic 1 level. When the initial voltage of the state capacitor is assigned to 0.5V, then the voltage at the V_2 node is more than that at the V_1 node, which in turn causes more current to flow through M55 (because $I_b = I_1 + I_2$). This causes the feedback voltage to drop and hence the state voltage also drops. This causes the output current to flow in the negative direction which indicates the logic 0 level. Figure 3.27 and Figure 3.28 show the transient response of the cell when the initial state voltages of the capacitor are at 4V and 0.5V respectively. The layout of the cell is shown in Figure 3.29.

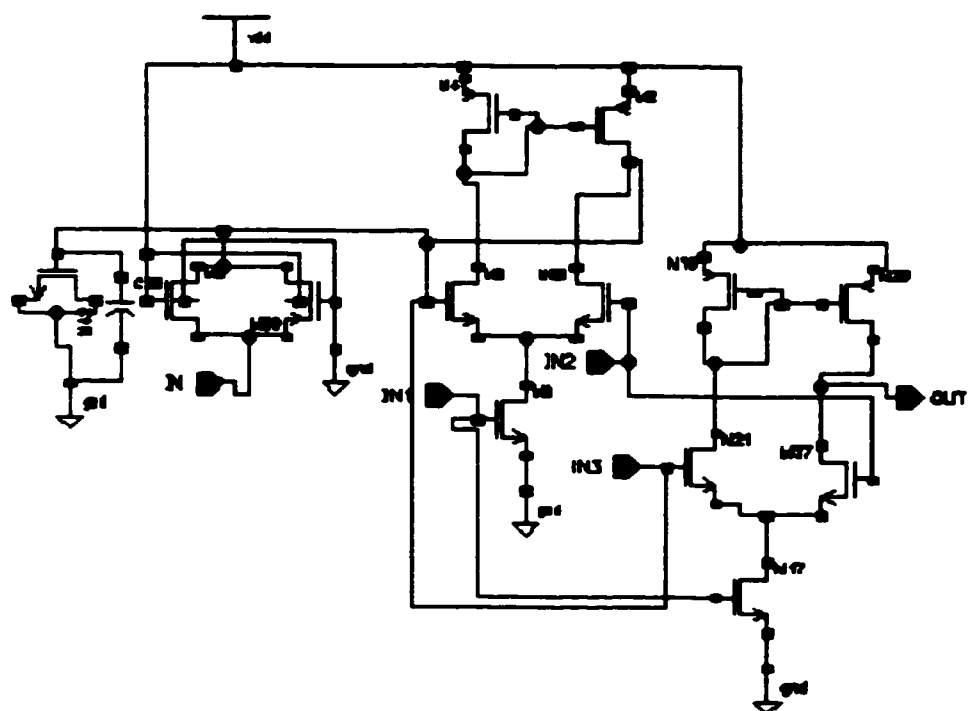
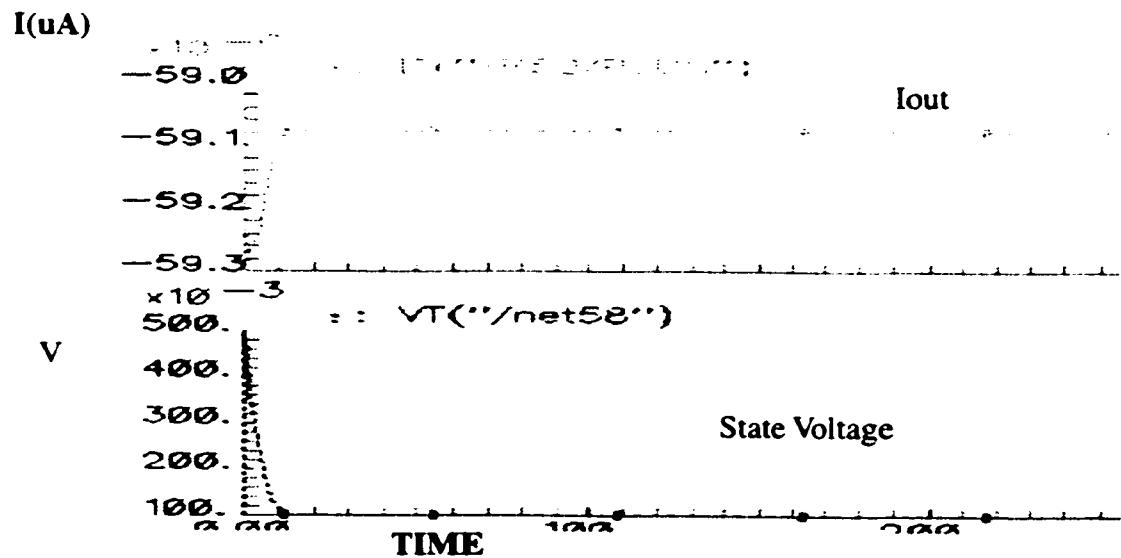
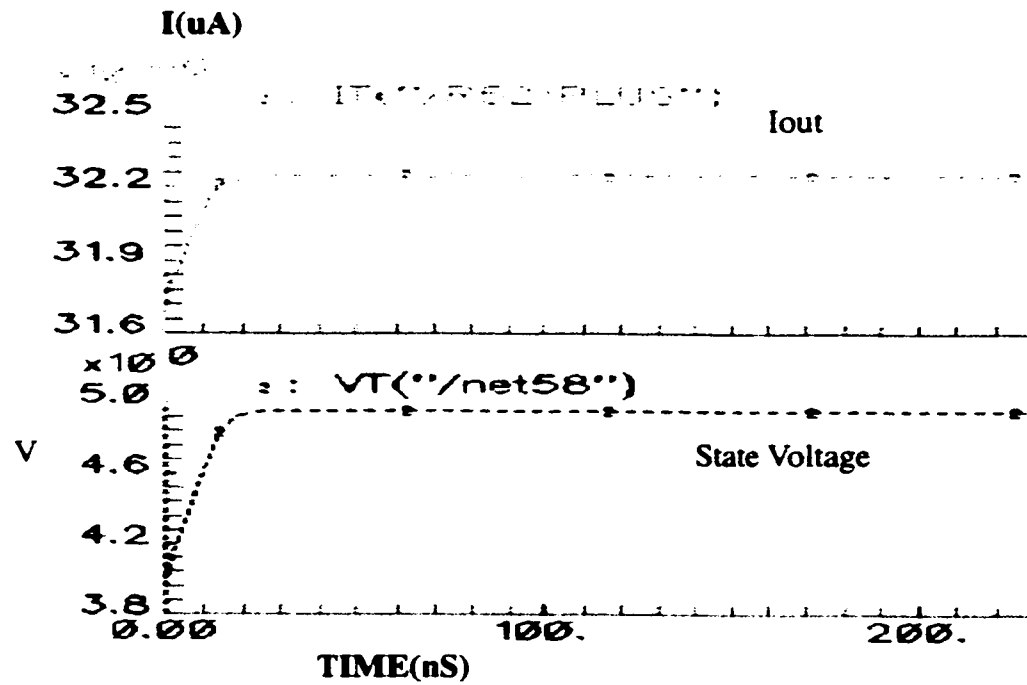


Figure 3.26 Basic cell circuit

Figure 3.27 Transient response when initial voltage is 4V**Figure 3.28 Transient response when initial voltage is 0.5V**

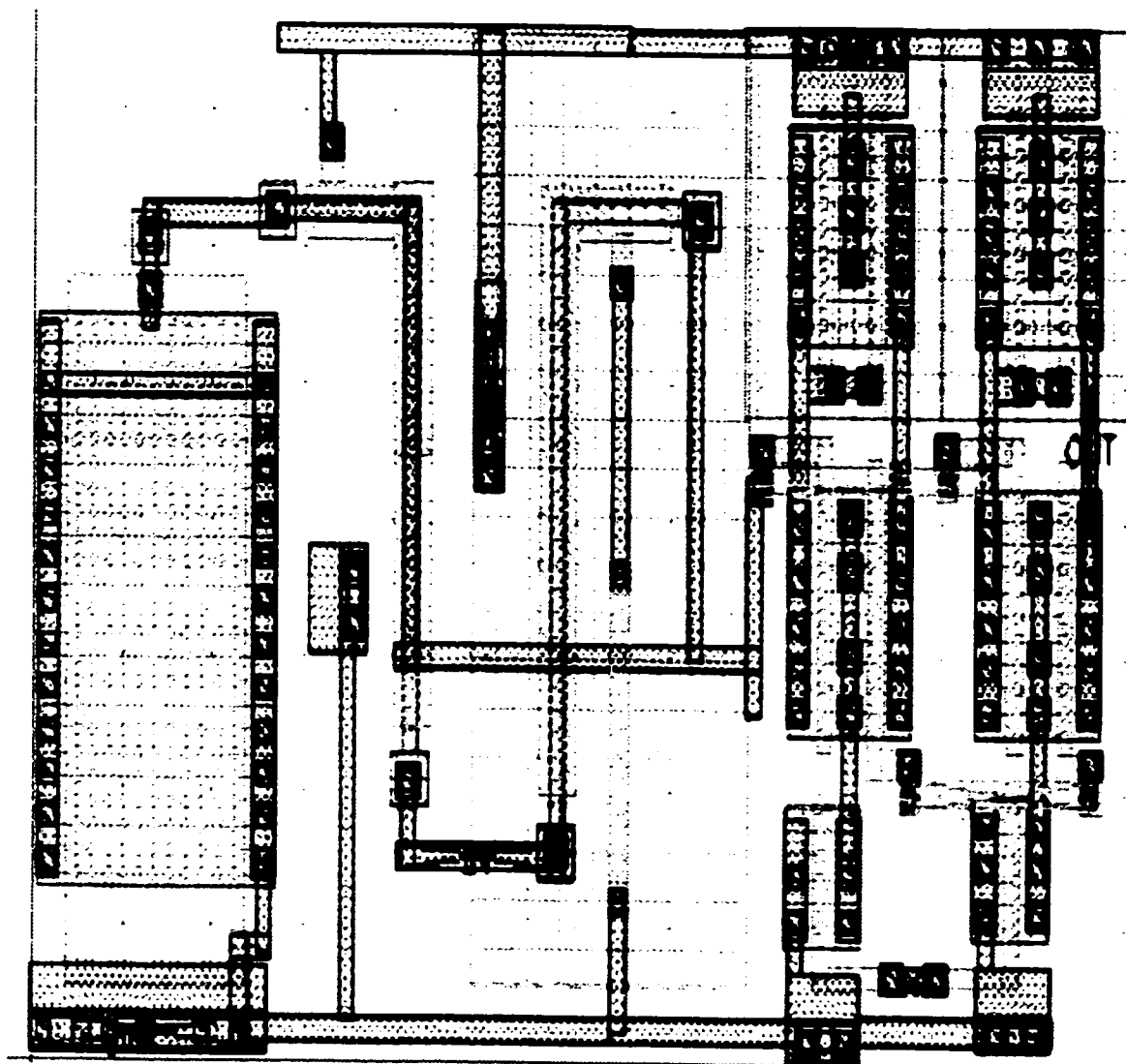


Figure 3.29 Layout of the basic CNN cell

3.6 Verification of Rule 2:

The algorithms that are used to implement binary arithmetic operations in cellular neural networks have been explained in chapter 2[20]. It may be recalled that binary addition can be carried out by applying two rules successively on the cells. The VLSI architecture to realize Rule 2 is shown in Figure 3.30. Rule 2 states that if there are any two adjacent cells in a column with the output voltage of the upper cell, $Y_u(t) = 1$ and the output voltage of the lower cell, $Y_d(t) = -1$, then the value of the upper cell will drop to -1 and, simultaneously, the value of the lower cell will rise to +1. Verification of the correctness of the circuit is carried out in the following manner.

Cell N1 is initialized with 4V across the state capacitor. The transient response of the cell is shown in Figure 3.31. As expected, positive current (logic '1') will flow through the output node. Cell N2 is initialized with 0.5V across its capacitor (logic '0') and the transient response is shown in Figure 3.32. We may realize the algorithm as follows:

- Add the output currents of the two cells. This is achieved by connecting the outputs into a common node (in this case we do not have to use a current adder circuit).
- Take the absolute value of the summed output by using an absolute value function circuit. Since the summed output is directly connected to the absolute circuit, one PMOS current mirror is eliminated from the structure.
- The corresponding output voltage can be attained by using a load at the output node of the absolute value circuit. This output voltage is used as the input for an inverter in order to obtain a steady voltage level. Since the state voltage of cell N1 increases and that of cell N2 decreases, the input of the inverter will be high and hence the output will be low. The output of the inverter is connected to the input of the hysteresis circuit which performs the switching of the logic levels. In order to guarantee correct switching, the gate-source voltage of the input transistors of the hysteresis circuit is initialized with 5V. This makes the output voltage of the hysteresis circuit to go from low to high voltage.
- The output of the hysteresis circuit is connected to the gates of two NMOS transistors, which turns the two transistors on. The drain of the upper transistor (M35) is connected to the state node of cell N1 and its source is connected to ground, whereas the drain of the bottom transistor (M33) is connected to V_{DD} and its source is connected to the state

node of cell N2. This setup causes the state voltage of cell N1 to drop and that of cell N2 to rise. So the logic level of N1 switches from logic 1 to logic 0, whereas logic level of N2 switches from logic 0 to logic 1, hence the Rule 2 is verified.

The simulation results of the Rule 2 verification is shown in Figure 3.31 and Figure 3.32.

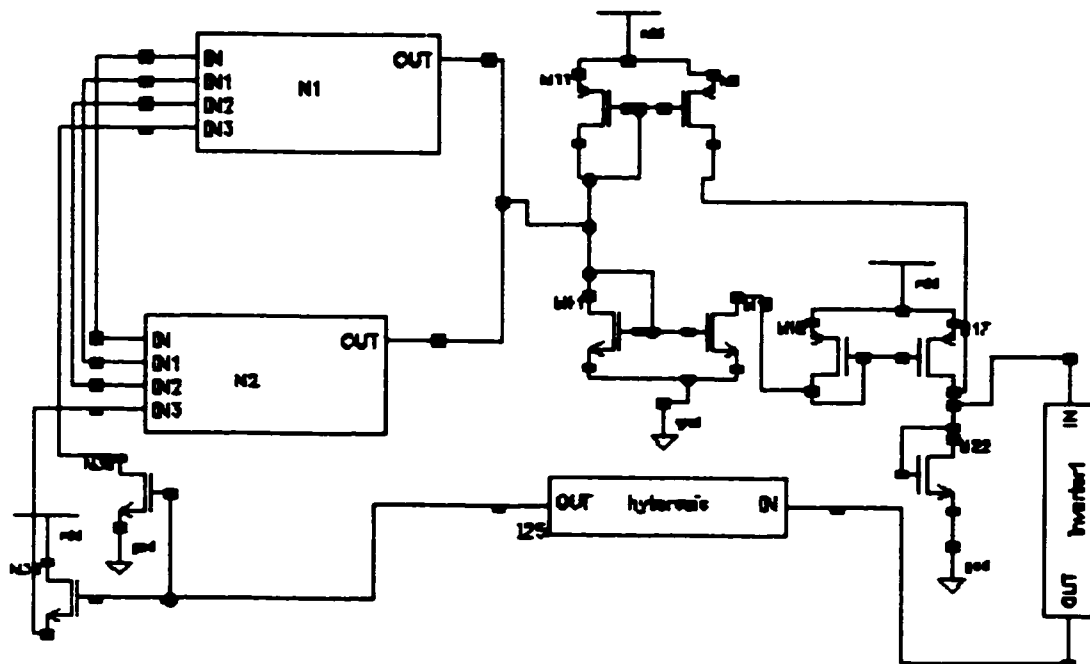
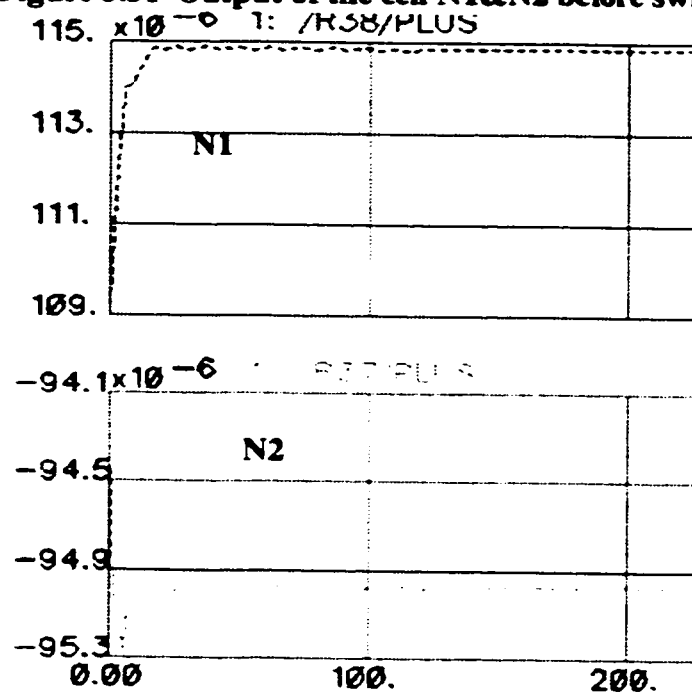


Figure 3.30 Verification of rule 2

Figure 3.31 Output of the cell N1&N2 before switching

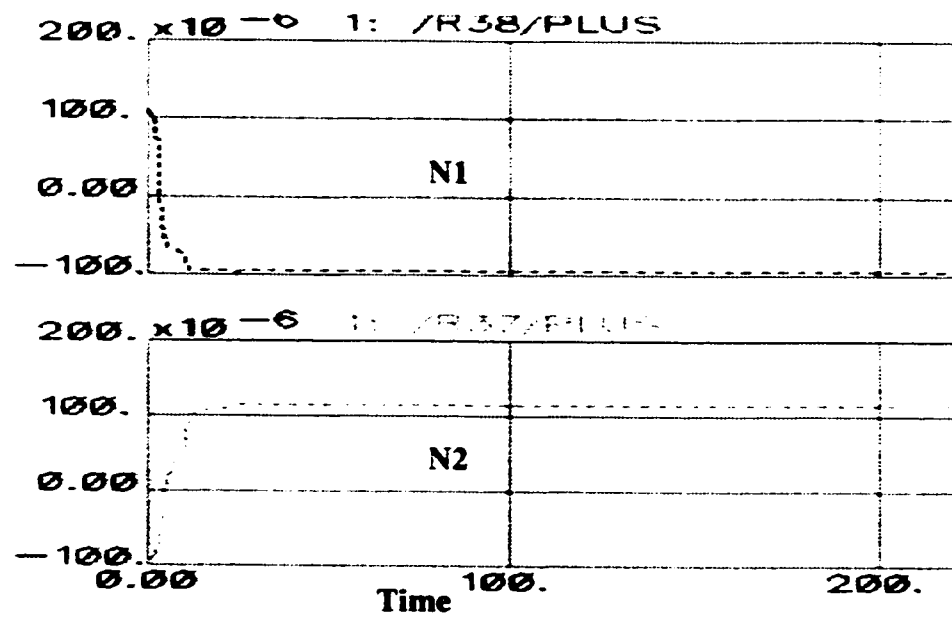


Figure 3.32 Output after switching

Chapter 4

SENSITIVITY ANALYSIS

4.1 Mismatches in MOS Transistors

The ideal behavior of MOS transistors is compromised by mismatches that are caused by variations at various levels of the fabrication process. These variations can occur in both fundamental process variables and transistor dimensions.

In general, there are three levels of statistical variations to consider in an IC process[43][44]:

- global variation between average chip values
- local variation within the chip
- mismatches reflecting the variations in component parameters between matched (adjacent) components on the same chip

The design of precision analog circuits requires an understanding of the matching behavior of the components. In the strong inversion region (in saturation), the drain current is given by:

$$I = \frac{\beta}{2}(V_{GS} - V_T)^2 \quad (4.1)$$

The variance in the drain current is given by

$$\frac{\gamma^2 I}{\bar{I}^2} = \frac{\gamma_\beta^2}{\beta^2} + 4 \frac{\gamma_{V_T}^2}{(V_{GS} - V_T)^2} \quad (4.2)$$

where

$\gamma_I, \gamma_\beta, \gamma_{V_T}$ = standard deviation of I , β , and V_T respectively

$\bar{I}, \bar{\beta}$, and \bar{V}_T = mean of I , β , and V_T

From Eqn. (4.2), it is evident that the dominant factor causing mismatch in drain current for low V_{GS} is the threshold voltage[45], but both β and V_T have equal contributions at near mid-rail values of V_{GS} .

4.1.1 Threshold Voltage Mismatch

The threshold voltage can be expressed as:

$$V_T = \Phi_{ms} + \Phi_b + \frac{Q_B}{C_{ox}} - \frac{Q_f}{C_{ox}} + q \frac{D_I}{C_{ox}} \quad (4.3)$$

where

Φ_{ms} = gate semiconductor work function difference

Φ_b = fermi potential in bulk

Q_B = depletion charge density

Q_f = fixed oxide charge density

D_I = threshold adjust implant

C_{ox} = gate oxide capacitance per unit area

Notice that Φ_{ms} and ϕ_b are constants and do not contribute to any mismatches. Also Q_f is much less than Q_B and hence can be neglected. Thus the variance is given by:

$$\gamma^2_{VT} = \frac{1}{LWC_{ox}}[q(Q_B + qDI)] + A_{ox}(Q_B^2 + q^2 + D^2I)] \quad (4.4)$$

We see that the standard deviation of mismatch is inversely proportional to the square root of the channel area. The non-uniform distribution of the dopant atoms in the bulk is a major contributor to the V_T mismatch[47].

4.1.2 Conductance Term Mismatch

The conductance term is given by:

$$\beta = \mu C_{ox} \frac{W}{L} \quad (4.5)$$

where μ is the channel mobility

The variance is given by,

$$\frac{\gamma^2 \beta}{\beta^2} = \frac{1}{LW}(A_\mu + A_{ox}) + \frac{\gamma^2 L}{L^2} + \frac{\gamma^2 W}{W^2}. \quad (4.6)$$

The mismatch in β due to edge variations is proportional to $\left(\frac{1}{L^2} + \frac{1}{W^2}\right)$.

A common contributing factor to the mismatches in V_T and in β is the variation in the gate-oxide capacitance[45][47].

4.2 Sensitivity Analysis

Sensitivity analysis indicates the extent to which a performance parameter varies with a small variation in a component / process parameter[43].

Normalized sensitivity S of a performance parameter F with respect to a component / process parameter P is given by:

$$S = \frac{P}{F} \left(\frac{\partial F}{\partial P} \right) \quad (4.7)$$

In order to get a better estimate of S , we must examine the behavior of F over a range of values of P ranging from P_{min} to P_{max} . Once the sensitivity of the performance to all the disturbances has been calculated, and the worst case directions are determined, the worst case for the performance may be calculated by adjusting all the disturbances in their worst case directions.

4.2.1 Sensitivity analysis of the basic CNN cell

In this analysis, the output current of the cell is the performance parameter. A sensitivity analysis is carried out on the performance parameter, F for different combinations of the threshold voltage V_T and the conductance parameter β (or K_p).

For a typical CMOSIS5 process, the values of V_T and K_p for NMOS and PMOS are shown in Table 4.1.

Table 4.1. Values of V_T and K_p for a typical CMOSIS5 process

Transistor	V_T	K_p
NMOS	0.6566V	1.9647E-4
PMOS	-0.9122V	4.8740E-5

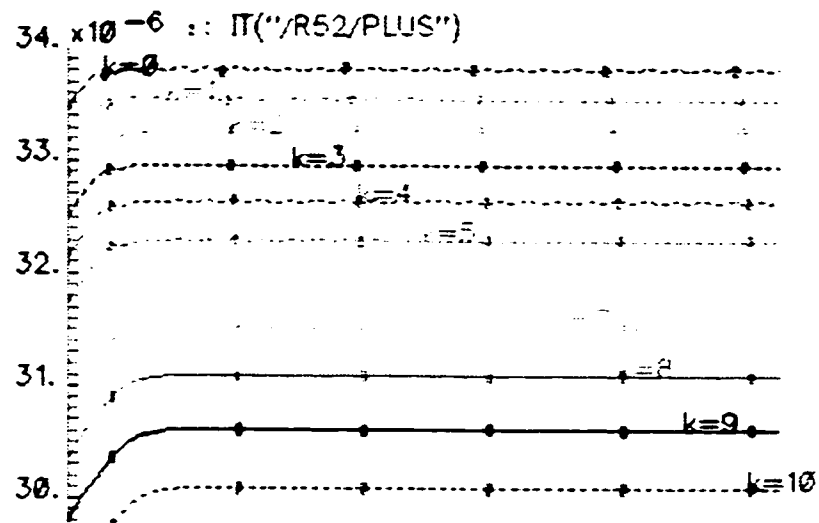
By applying a 10% tolerance of V_T and K_p on the basic cell, the variations in the output current are determined for the following different combinations of the process parameters.

- Varying V_T of NMOS only
- Varying V_T of both PMOS and NMOS
- Varying K_p of NMOS
- Varying K_p of both NMOS and PMOS
- Varying K_p and V_T of both NMOS and PMOS

Figures 4.1 to 4.5 show the simulation results for the above mentioned combinations. The resulting output currents due to variations of the process parameters are measured and recorded in Table 4.2. The expression $k=0$ indicates the minimum tolerance and $k=10$ indicates the maximum tolerance. The deviation of the output current from its normal condition to the worst case are calculated for the above cases and are provided in Table 4.3, 4.4 and 4.5.

Table 4.2. Measured output current from different combinations

	Iout(k=0) U	Iout(k=10) U
VTn	33.81	30.89
VTn & VTp	33.84	33.00
KPn	31.69	32.68
KPn & KPp	31.76	32.69
VT & KP	32.92	30.97

**Figure 4.1 Varying V_T of NMOS**

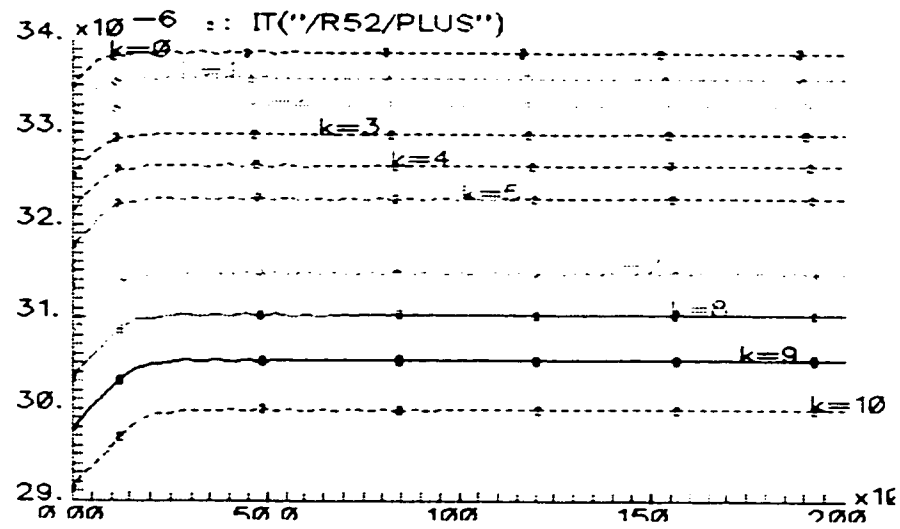


Figure 4.2 Varying V_T of both NMOS and PMOS

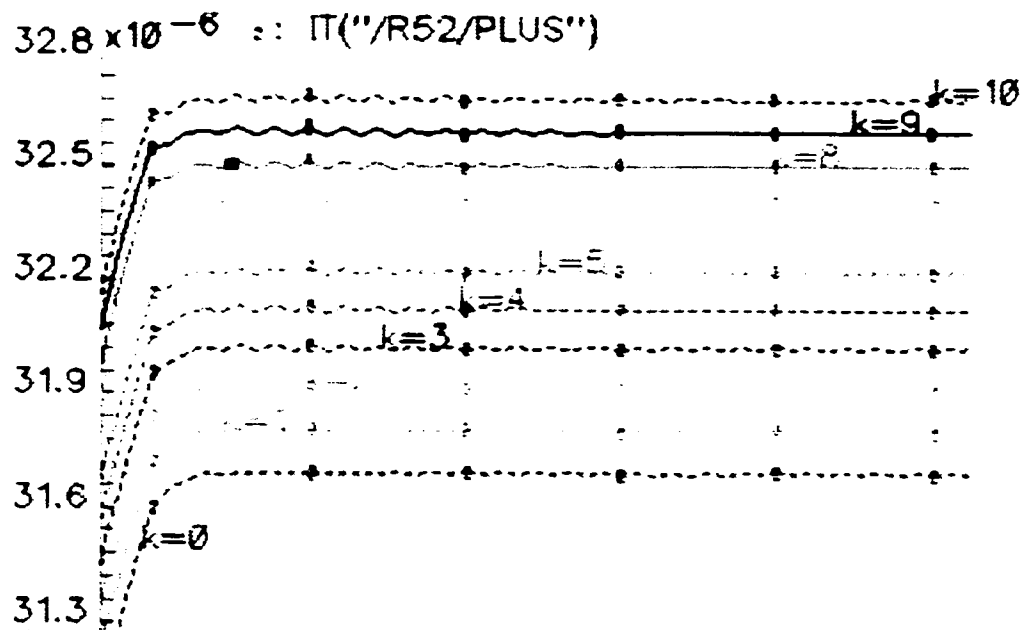


Figure 4.3 Varying K_p of NMOS

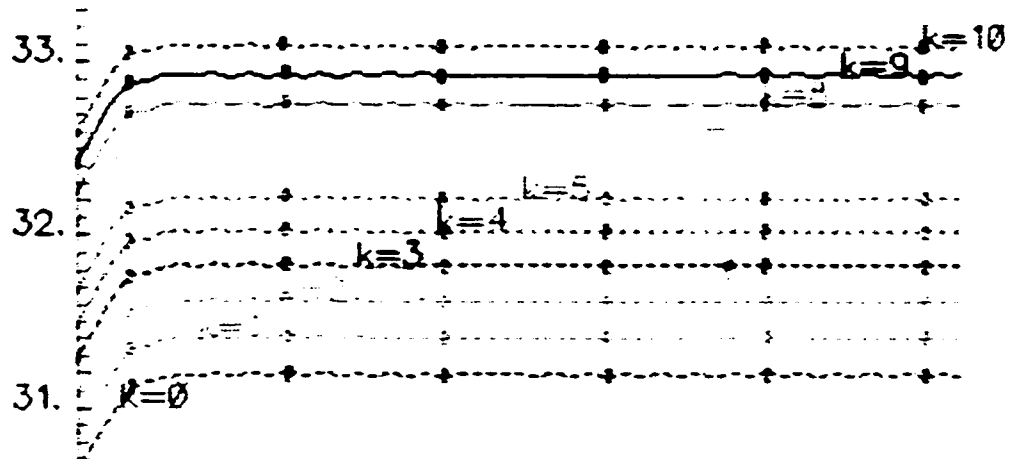


Figure 4.4 Varying K_p of both NMOS and PMOS

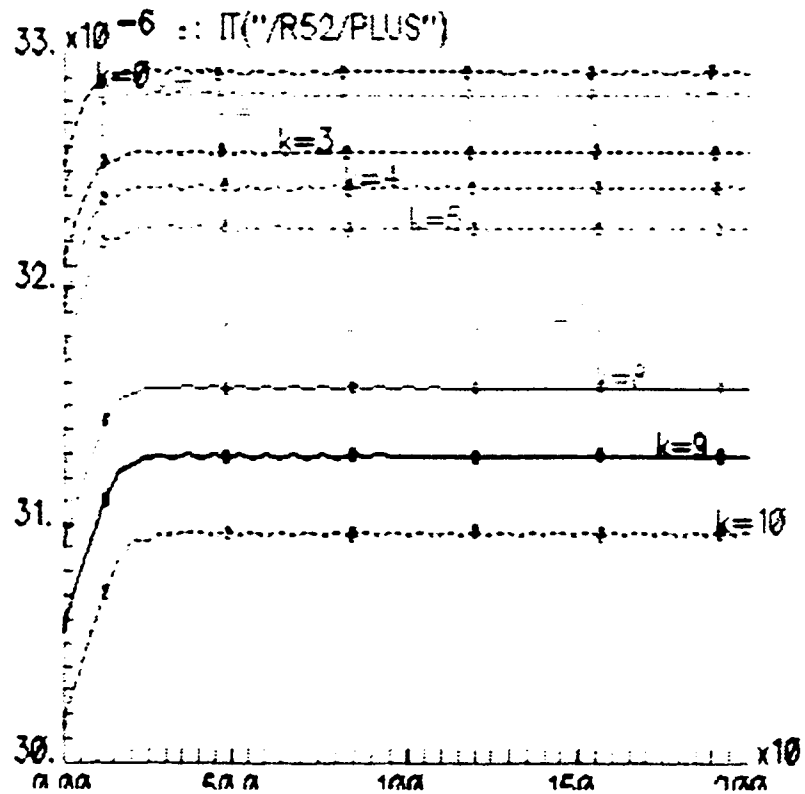


Figure 4.5 Varying K_p and V_T of both NMOS and PMOS

Table 4.3. Influence of V_T

	Vt1	Vt	Vt2	Vt-Vt1	Vt-Vt2
NMOS	33.81 U	32.2 U	30.09 U	-1.61 U	2.11 U
N&P MOS	33.84 U	32.2 U	33.00 U	-1.64 U	0.8 U
Vt1:VT(k=0), Vt2:VT(k=10), Vt: VT(normal), N&P MOS:NMOS & PMOS					

Table 4.4. Influence of K_p

	Kp1	Kp	Kp2	Kp-Kp1	Kp-Kp2
NMOS	31.69 U	32.2 U	32.68 U	0.51 U	-0.48 U
N&P MOS	31.76 U	32.2 U	32.69 U	0.44 U	-0.49 U
Kp1:KP(k=0), Kp2:KP(k=10), Kp: KP(normal), N&P MOS:NMOS & PMOS					

Table 4.5. Influence of V_T and K_p

	R1	R	R2	R-R1	R-R2
N&P MOS	32.92 U	32.2 U	30.97 U	-0.72 U	1.23 U
R1:KP&VT(k=0), R2:KP&VT(k=10), R: KP&VT(normal), N&P MOS:NMOS & PMOS					

4.2.2 Observations

The following observations are made when performing a sensitivity analysis on the output current of a basic CNN cell. The parameters are varied by applying a $\pm 10\%$ tolerance

1. The circuit is more sensitive to variations in threshold voltage than to variations in the conductance parameter.
2. The sensitivity of the output current when V_T and K_P are separately applied on the circuit is different from the combined influence of V_T and K_P . This shows a common contributing factor to the mismatches in V_T and K_P , and is due to the variation in the gate-oxide capacitance of the MOS transistor.
3. The sensitivity of the output current of the basic cell due to parameter disturbances are monotonic and nearly linear.
4. A 10% tolerance on process parameters yields less than a 7% variation in circuit performance.

4.3 Summary

This chapter described the sensitivity analysis on a basic CNN cell. The analysis is conducted by identifying the process parameters that cause mismatch in transistor behavior, i.e., threshold voltage (V_t) and transconductance term parameter (β or K_p) and applying 10% tolerance on these parameters to monitor the influence of these disturbance on the performance of the circuit. The simulations are performed by choosing five different combinations of V_t and K_p of both NMOS and PMOS transistors. The P_{min} has been chosen as a starting value and incremented with a constant step. Parametric simulation is employed to conduct the analysis. It has been observed that the sensitivity of the output current of the basic cell due to parameter variations is monotonic and nearly linear. Also the 10% tolerance on process parameters yields less than a 7% variation in the circuit performance.

Chapter 5

CONCLUSIONS

This thesis has investigated different aspects of designing analog library cells for Cellular Neural Network applications. A sensitivity analysis for the worst case design has also been conducted.

The silicon implementation for realizing the building blocks for image processing and binary arithmetic operations has been the main thrust of this thesis. The building blocks for image processing include: an operational transconductance amplifier (or nonlinear V/I converter) with programming capabilities; a linear tunable V/I converter; a cell resistor and a cell capacitor. The circuits have been designed using state of the art techniques and the functionalities of these circuits have been verified by performing Design Rule Check (DRC), Layout Versus Schematic (LVS), and post layout simulations. The parasitics extracted from the layout have a minor influence on the performance of the circuits.

The building blocks for binary arithmetic operations include: an absolute value function circuit; a current comparator; a hysteresis circuit; a current adder; and a current subtractor. Since the network operates in current mode, all the above mentioned circuits are designed using simple current mirrors (both NMOS and PMOS),

and differential pairs. The operations of these circuits have also been verified by conducting DRC, LVS, and post layout simulations.

The speed of the basic CNN cell is calculated by applying different state voltages across the capacitor and observing the transient response of the circuit. The speed is approximated as the inverse of the time difference between starting the circuit evaluation and the time at which the output becomes stable. It is found that the basic CNN cell operates at 40MHz.

The CNN cell employs a linear resistor to realize the loss term. In order to realize a cell resistor, we use an NMOS and a PMOS transistor operating in the linear region; this occupies a relatively large area in the layout. In order to test the robustness of the design to a more area efficient resistor, we replaced the linear resistor with a nonlinear resistor built by connecting the gate of an NMOS transistor to its drain and operating in the saturation region. Surprisingly, the influence of the nonlinear resistor to the performance of the circuit is found to be negligible at the cell level. However this nonlinear structure may not exhibit similar behavior in a complete CNN array due to possible effects of the potential race condition. This situation arises due to fluctuating RC time constants of cells in a typical operating neighbourhood.

A sensitivity analysis was also carried out on the basic CNN cell in order to determine its tolerance with regard to the variations of the process parameters. The simulations are performed by choosing the threshold voltage (V_T) and the conductance parameter (K_P) of the MOS transistor as the potential component parameters and varying these parameters by 10% tolerance. The influence of these variations on the output has been monitored. It has been observed that the circuit is more sensitive to variations in V_T than in K_P . We also notice that the sensitivity of the output, when V_T and K_P separately applied to the circuit, is different from the combined influence of V_T and K_P . This shows a correlation between the mismatches in V_T and K_P ; we conclude that this is due to the variation in the

gate-oxide capacitance of the MOS transistor. The sensitivity of the output due to parameter disturbances is monotonic and nearly linear. Furthermore, a 10% tolerance on the process parameter has yielded less than 7% variations on the circuit performance, this shows that the basic CNN cell circuit is reasonably robust.

REFERENCES

- [1] J. J Hopfield, "Neural networks and physical systems with emergent computational abilities," Proc. Natl. Acad. Sci. USA., vol. 79, pp. 2554-2558, 1982.
- [2] J. J. Hopfield and D. W. Tank, "'Neural' computation of decisions in optimization problems," Biol. Cybern., vol. 52, pp. 141-152, 1985
- [3] L. O. Chua and L. Yang, "Cellular neural networks: Theory and applications," IEEE Trans. Circuits Syst., vol. 35, pp. 1257-1290, Oct. 1988.
- [4] L. O. Chua and T. Roska, "The CNN paradigm", IEEE Trans. Circuits and Systems-1: Fundamental Theory and applications, vol. 40, No. 3, March 1993.
- [5] J. A. Nossek, T. Roska, G. Seler, and L. O. Chua, "Cellular neural networks: Theory and Circuit Design", pp. 94-115
- [6] T. Roska, and L. O. Chua, "Cellular neural networks with nonlinear and delay-type template elements and nonuniform grids," Int. J. Circuit Theory and Applications., vol. 20, pp. 469-482, 1992.
- [7] Peter Kinget and Michel S. J. Steyaert, "A programmable Analog cellular Neural Network CMOS chip for High speed Image processing", IEEE J. of solid-state circuits, vol. 30, pp. 235-243, March 1995.
- [8] T. Matsumoto, L O. Chua, and R. Furukawa, "CNN Cloning template: hole-filler," IEEE Trans. Circuits Syst., vol. 37, pp. 635-638, May 1990.
- [9] P. Kinget and M. steyaert, "Analogue CMOS VLSI implementation of cellular neural networks with continuously programmable templates", Proc. IEEE Int. Symp. Circuits Syst., vol. 6, pp. 367-370, May 1994.
- [10] J. M. Cruz and L. O. Chua, "A CNN chip for connected component detection," IEEE Trans. Circuits Syst., vol. 38, pp. 810-817, July 1991.
- [11] A. Rodriguez- Vasquez, S. Espejo, R. Dominguez-Castro, J. L Huertas, and E. Sanchez- Sinencio, "A Current-Mode techniques for the implementation of continuous- and discrete time cellular neural networks," IEEE Trans. Circuits Syst. II, vol. 40, pp. 132-145, March 1993.

-
- [12] S. Espejo, A. Rodriguez-Vasquez, R. Dominguez-Castro, J. L. Huertas. and E. Sanchez-Sinencio, "Smart-pixel cellular neural networks in analog current -mode CMOS technology," *IEEE J. Solid-state Circuits*, vol. 29, no. 8, pp. 895-905, August 1994.
 - [13] P. Kinget and M. Steyaert, "Evaluation of CNN template robustness towards VLSI implementation," *Proc. 3rd IEEE Int. Workshop Cellular Neural Networks Application*, (CNNA94), pp. 381-386, December 1994.
 - [14] F. Lobato-Lopez and J. Silva_Martinez and E. Sanchez-Sinencio, "A Very Fast CMOS Artificial Cellular Neural Networks," *Proceedings of the IEEE Int. Symposium on Circuits and Systems*, vol. 3, pp. 418-421, May 1996.
 - [15] A. Rodriguez, R. Dominguez-Castro, and S. Espejo, "Design of CNN Universal Chips: Trends and Obstacles," *CNNA94*, December 1994.
 - [16] *Neural Information Processing and VLSI*.
 - [17] T. Roska, L. O. Chua, "The CNN Universal Machine- An Analogic Array Computers," *IEEE Trans. Circuits and Systems*, vol. 40, pp. 163-173, March 1993.
 - [18] A. Sani, S. Grsffi, G. Masetti, G. Setti, "Design of CMOS Cellular Neural Networks Operating at Several Supply Voltages," *CNNA94*, pp. 363-369, December 1994.
 - [19] A. Passio, A. dawidziak, K. Halonen, and V. Porra, "Digitally Controllable Weights in Current Mode Cellular Neural Networks," *CNNA-94*, pp. 73, December 1994.
 - [20] Saeid Sadehi-Emamchaie, G. A. Jullien and W. C. Miller, "Arithmetic Arrays Using Cellular Neural Networks." *31st Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, Ca, Nov. 2-5, 1997.
 - [21] C. A. Mead, *Analog VLSI and Neural Systems*. Reading, MA: Addison Wesley, 1989.
 - [22] R. D. Reed and R. L. Geiger, "A Multiple Input OTA Circuit for Neural Networks," *IEEE Trans. Circuits Syst.*, vol. CAS-36, May 1989.
 - [23] Mohammed Ismail and Terrifiez, *Analog VLSI Signal and Information Processing*.
 - [24] P. Allen, D. Holberg, *CMOS Analog Circuit Design*, Hol, Rinehart, and Winston Inc., 1987.
 - [25] G. C. Cardarilli, R. Lojacono, M. Salerio, F. Sargeni, "VLSI Implementation of Programmable cellular neural networks," *ICANN-92*, Brighton, U.K, September 1992.
 - [26] Kenneth R. Laker and Willy. M. C. Sansen, *Design of Analog Integrated Circuits and Systems*.
-

-
- [27] K. Bult and H. Walling, "A Class of Analog CMOS Circuits Based on the Square-Law Characteristic of an MOS Transistor in Saturation," *IEEE J. Solid-State Circuits*, vol. 22, pp. 357-364, June 1987.
 - [28] E. Sanchez-Sinencio, J. Ramirez-Angulo, B. Linares-Barranco, A. Rodriguez-Vazquez, "Operational Transconductance Amplifier-Based Nonlinear Function Syntheses," *IEEE J. Solid-State Circuits*, vol. 24, pp. 1576-1586, December 1989.
 - [29] E. A. Vittoz, "The design of High Performance Analog Circuits on a Digital CMOS Chips," *IEEE J. Solid-State Circuits*, vol. 20, pp. 657-665.
 - [30] N. I. Khachab and M. Ismail, "A Nonlinear CMOS Analog Cell for VLSI Signal and Information Processing," *IEEE J. Solid-State Circuits*, vol. 26, pp. 1689-1698, November 1991.
 - [31] S. C. Huang and M. Ismail, "Linear Tunable COMFET Transconductor," *Electron. Lett.*, vol. 29, pp. 459-461, March 1993.
 - [32] Cheng, Toumazou, "Linear Composite MOSFETs(COMFETs)," *Electron. Lett.*, pp. 1802-1804, Sept. 1991.
 - [33] E. Seevinck, and R. F. Wassenaar, "A Versatile CMOS Linear Transconductor/Square-Law Function Circuit," *IEEE J. Solid-State Circuits*, pp. 367-377, 1987.
 - [34] J. M. Cruz and L. O. Chua, "A CMOS IC Nonlinear Resistor for Chua's Circuit," *IEEE Trans. Circuits Syst-Fundamental Theory and Applications*, vol. 39, pp. 985-995, December 1992.
 - [35] Y. Tsividis, *Operation and Modeling of the MOS Transistor*, McGraw-Hill, 1986.
 - [36] F. Euno, T. Inoue, and Y. Shirai, "A Maximum and Minimum Circuit with Multiple Inputs in Current Mode," *The Trans. of the IEICE*, vol. E-70, pp. 392-395, April 1987.
 - [37] A. T. K. Tang and C. Toumazou, "High Performance CMOS Current Comparator," *Electron. Lett.* vol. 30, 6th. January 1994.
 - [38] G. Palmisano and G. Palumbo, "Offset Compensation Technique for CMOS Current Comparators," *Electron. Lett.* vol. 30, pp. 852-854, 26-Th. May 1994.
 - [39] I. M. Filanovsky and H. Baltes, "CMOS Schmitt Trigger Design," *IEEE Trans. Circuits Syst- Fundamental Theory and Applications*, vol. 41, pp. 46-49, January 1994.
 - [40] M. J. S Smith, "On The Circuit Analysis of the Schmitt Trigger," *IEEE J. Solid-State Circuits*, vol. 23, pp. 292-294, Feb. 1988.
-

-
- [41] B. L. Dokic, "CMOS Schmitt Triggers," IEE Proc. G. Electron. Circuits and Syst. pp. 197-202, 1984.
 - [42] Z. Wang and W. Guggenbuhl, "Novel CMOS Current Schmitt Trigger," Electron Lett., vol. 24, pp. 1514-1516, 24-Th. November 1988.
 - [43] R. Spence and R.S. Soin, Tolerance Design of Electronic Circuits. Reading, MA: Addison Wesley, 1988.
 - [44] S. R Nassif, A Strojwas and S. W. Director, "A Method for Worst Case Analysis of Integrated Circuits," IEEE Trans. Computer-Aided Design, vol. CAD-5, pp. 104-113, January 1986.
 - [45] S. M Sze, Semiconductor Device Physics and Technology. New York: John Wiley & Sons, 1987.
 - [46] P. R. Gary, "Basic MOS Operational Amplifiers Design- An Overview," in Analog MOS Integrated Circuits, P. R Gary, D. A. Hodges and R. W. Brodersen, Eds., IEEE Press, 1980.
 - [47] K. R. Lakshikumar, R. A. Hadaway and M. A. Copeland, "Characterization and Modeling of Mismatch in MOS Transistors for Precision Analog Design," IEEE J. Solid-State Circuits, vol. Sc-21, pp. 1057-1066, December 1986.

Appendix A

SPICE netlist of OTA circuit

```
# vdc Instance V45 = spectreS device v45 v45 g1 0 \source type=\dc dc=5.0

# vdc Instance V64 = spectreS device v64 v64 net58 0 \source type=\dc dc=vdc1

# vdc Instance V63 = spectreS device v63 v63 net68 0 \source type=\dc dc=2.5

# nfet3 Instance M21 = spectreS device m21 m21 net54 net58 net57 0 \CMOSN
region=\triode w=9.2e-6 l=800e-9 as=1e-6*9.2e-6 ad=1e-6*9.2e-6 ps=2e-6+2.0*9.2e-6
pd=2e-6+2.0*9.2e-6 nrd=1e-6/9.2e-6 & nrs=1e-6/9.2e-6 m=1.0

# nfet3 Instance M17 = spectreS device m17 m17 net57 net66 0 0 \CMOSN region=\tri-
ode w=5.2e-6 l=800e-9 as=1e-6*5.2e-6 ad=1e-6*5.2e-6 ps=2e-6+2.0*5.2e-6 pd=2e-
6+2.0*5.2e-6 nrd=1e-6/5.2e-6 & nrs=1e-6/5.2e-6 m=1.0

# nfet3 Instance M57 = spectreS device m57 m57 out net68 net57 0 \CMOSN
region=\triode w=9.2e-6 l=800e-9 as=1e-6*9.2e-6 ad=1e-6*9.2e-6 ps=2e-6+2.0*9.2e-6
pd=2e-6+2.0*9.2e-6 nrd=1e-6/9.2e-6 & nrs=1e-6/9.2e-6 m=1.0

# pfet3 Instance M19 = spectreS device m19 m19 net54 net54 g1 g1 \CMOSP
region=\triode w=8e-6 l=800e-9 as=1e-6*8e-6 & ad=1e-6*8e-6 ps=2e-6+2.0*8e-6 pd=2e-
6+2.0*8e-6 nrd=1e-6/8e-6 nrs=1e-6/8e-6 & m=1.0

# pfet3 Instance M20 = spectreS device m20 m20 out net54 g1 g1 \CMOSP region=\tri-
ode w=8e-6 l=800e-9 as=1e-6*8e-6 & ad=1e-6*8e-6 ps=2e-6+2.0*8e-6 pd=2e-6+2.0*8e-
6 nrd=1e-6/8e-6 nrs=1e-6/8e-6 & m=1.0
```

Appendix B

SPICE netlist of the linear V/I converter

```
# resistor Instance R18 = spectreS device r18 r18 0 net1 \resistor r=25e3 m=1.0

# resistor Instance R17 = spectreS device r17 r17 net1 g1 \resistor r=5e3 m=1.0

# resistor Instance R16 = spectreS device r16 r16 net39 net1 \resistor r=3e3 m=1.0

# vdc Instance V25 = spectreS device v25 v25 net31 0 \source type=\dc dc=2.5

# vdc Instance V24 = spectreS device v24 v24 net7 0 \source type=\dc dc=vdc2

# vdc Instance V15 = spectreS device v15 v15 net25 0 \source type=\dc dc=vdc1

# vdc Instance V14 = spectreS device v14 v14 g1 0 \source type=\dc dc=5.0

# pfet3 Instance M12 = spectreS device m12 m12 net27 net27 g1 g1 \CMOSP
region=\triode w=80e-6 l=4e-6 as=1e-6*80e-6 & ad=1e-6*80e-6 ps=2e-6+2.0*80e-6
pd=2e-6+2.0*80e-6 nrd=1e-6/80e-6 nrs=1e-6/& 80e-6 m=1.0

# pfet3 Instance M9 = spectreS device m9 m9 net39 net27 g1 g1 \CMOSP region=\triode
w=80e-6 l=4e-6 as=1e-6*80e-6 & ad=1e-6*80e-6 ps=2e-6+2.0*80e-6 pd=2e-6+2.0*80e-
6 nrd=1e-6/80e-6 nrs=1e-6/& 80e-6 m=1.0

# pfet3 Instance M8 = spectreS device m8 m8 0 net34 net23 g1 \CMOSP region=\triode
w=80e-6 l=2e-6 as=1e-6*80e-6 & ad=1e-6*80e-6 ps=2e-6+2.0*80e-6 pd=2e-6+2.0*80e-
6 nrd=1e-6/80e-6 nrs=1e-6/& 80e-6 m=1.0
```

```
# pfet3 Instance M10 = spectreS device m10 m10 0 net28 net20 g1 \CMOSP region=\tri-
ode w=80e-6 l=2e-6 as=1e-6*80e-6 & ad=1e-6*80e-6 ps=2e-6+2.0*80e-6 pd=2e-
6+2.0*80e-6 nrd=1e-6/80e-6 nrs=1e-6/& 80e-6 m=1.0
```

```
# nfet3 Instance M13 = spectreS device m13 m13 net27 net25 net28 0 \CMOSN
region=\triode w=10e-6 l=20e-6 as=1e-6*& 10e-6 ad=1e-6*10e-6 ps=2e-6+2.0*10e-6
pd=2e-6+2.0*10e-6 nrd=1e-6/10e-6 & nrs=1e-6/10e-6 m=6.0
```

```
# nfet3 Instance M7 = spectreS device m7 m7 net39 net31 net34 0 \CMOSN region=\tri-
ode w=10e-6 l=20e-6 as=1e-6*10e-& 6 ad=1e-6*10e-6 ps=2e-6+2.0*10e-6 pd=2e-
6+2.0*10e-6 nrd=1e-6/10e-6 nrs=1e-& 6/10e-6 m=6.0
```

```
# nfet3 Instance M6 = spectreS device m6 m6 net34 net34 0 0 \CMOSN region=\triode
w=10e-6 l=20e-6 as=1e-6*10e-6 & ad=1e-6*10e-6 ps=2e-6+2.0*10e-6 pd=2e-
6+2.0*10e-6 nrd=1e-6/10e-6 nrs=1e-6/& 10e-6 m=6.0
```

```
# nfet3 Instance M5 = spectreS device m5 m5 net28 net28 0 0 \CMOSN region=\triode
w=10e-6 l=20e-6 as=1e-6*10e-6 & ad=1e-6*10e-6 ps=2e-6+2.0*10e-6 pd=2e-
6+2.0*10e-6 nrd=1e-6/10e-6 nrs=1e-6/& 10e-6 m=6.0
```

```
# nfet3 Instance M3 = spectreS device m3 m3 net27 net7 net23 0 \CMOSN region=\tri-
ode w=10e-6 l=20e-6 as=1e-6*10e-& 6 ad=1e-6*10e-6 ps=2e-6+2.0*10e-6 pd=2e-
6+2.0*10e-6 nrd=1e-6/10e-6 nrs=1e-& 6/10e-6 m=6.0
```

```
# nfet3 Instance M1 = spectreS device m1 m1 net39 net7 net20 0 \CMOSN region=\tri-
ode w=10e-6 l=20e-6 as=1e-6*10e-& 6 ad=1e-6*10e-6 ps=2e-6+2.0*10e-6 pd=2e-
6+2.0*10e-6 nrd=1e-6/10e-6 nrs=1e-& 6/10e-6 m=6.0
```


Appendix C

SPICE netlist of programmable nonlinear V/I converter

```
# ota2 Instance I27 = spectreS device xi27 xi27 net40 net46 net48 net57 net58
cnllib1_ota2_schematic

# res Instance R24 = spectreS device r24 r24 net37 net59 \resistor r=1e3 m=1.0

# res Instance R13 = spectreS device r13 r13 net37 0 \resistor r=100e3 m=1.0

# res Instance R12 = spectreS device r12 r12 net57 net37 \resistor r=10e3 m=1.0

# vdc Instance V29 = spectreS device v29 v29 net58 net39 \vsource type=\dc dc=0.0

# vdc Instance V28 = spectreS device v28 v28 net67 net58 \vsource type=\dc dc=0.0

# vdc Instance V20 = spectreS device v20 v20 net40 0 \vsource type=\dc dc=vdc1

# vdc Instance V21 = spectreS device v21 v21 net42 0 \vsource type=\dc dc=vdc2

# vdc Instance V22 = spectreS device v22 v22 net56 0 \vsource type=\dc dc=vdc3

# vdc Instance V8 = spectreS device v8 v8 net46 0 \vsource type=\dc dc=2.5

# vdc Instance V7 = spectreS device v7 v7 net48 0 \vsource type=\dc dc=900e-3

# vdc Instance V4 = spectreS device v4 v4 net57 0 \vsource type=\dc dc=5.0
```

```

# gilbert1 Instance I2 = spectreS device xi2 xi2 net57 net67 net42 net59
cnllib1_gilbert1_schematic

# terminal mapping: IN1 = IN1 #          IN2 = IN2 #          IN3 = IN3 #
IN4 = IN4 #          OUT1 = OUT1

.SUBCKT &1 IN1 IN2 IN3 IN4 OUT1

# nmos Instance M37 = spectreS device m37 m37 net21 in1 net39 net39 \mnch_0p8
region=\sat w=5e-6 l=800e-9 m=1.0

# pmos Instance M41 = spectreS device m41 m41 net21 net21 in4 in4 \mpch_0p8
region=\sat w=5e-6 l=800e-9 m=1.0

# pmos Instance M40 = spectreS device m40 m40 out1 net21 in4 in4 \mpch_0p8
region=\sat w=5e-6 l=800e-9 m=1.0

# nmos Instance M38 = spectreS device m38 m38 out1 in2 net39 net39 \mnch_0p8
region=\sat w=5e-6 l=800e-9 m=1.0

# nmos Instance M39 = spectreS device m39 m39 net39 in3 0 0 \mnch_0p8 region=\sat
w=4e-6 l=800e-9 m=1.0

# End of subcircuit definition. .ENDS &1 #terminal mapping: IN1 = IN1 # IN2 = IN2 #
IN3 = IN3 #          OUT = OUT

.SUBCKT &1 IN1 IN2 IN3 OUT

# pmos Instance M15 = spectreS device m15 m15 net42 in3 in2 in2 \mpch_0p8
region=\triode w=60e-6 l=10e-6 m=1.0

# pmos Instance M16 = spectreS device m16 m16 net5 0 in2 in2 \mpch_0p8 region=\tri-
ode w=60e-6 l=10e-6 m=1.0

```

```

# pmos Instance M4 = spectreS device m4 m4 net21 net21 net5 net5 \mpch_0p8
region=\triode w=60e-6 l=10e-6 m=1.0 &

# pmos Instance M2 = spectreS device m2 m2 net18 net21 net42 net42 \mpch_0p8
region=\triode w=60e-6 l=10e-6 & m=1.0

# nmos Instance M17 = spectreS device m17 m17 out net18 0 0 \mnch_0p8 region=\tri-
ode w=60e-6 l=10e-6 m=1.0

# nmos Instance M18 = spectreS device m18 m18 in1 in1 0 0 \mnch_0p8 region=\triode
w=60e-6 l=10e-6 m=1.0

# nmos Instance M3 = spectreS device m3 m3 net18 net18 0 0 \mnch_0p8 region=\triode
w=60e-6 l=10e-6 m=1.0

# nmos Instance M1 = spectreS device m1 m1 net21 in1 0 0 \mnch_0p8 region=\triode
w=60e-6 l=10e-6 m=1.0 # End of subcircuit definition. .ENDS &1 # terminal mapping:
IN1 = IN1 #          IN2 = IN2 #          IN3 = IN3 #          OUT = OUT

.SUBCKT &1 IN1 IN2 IN3 OUT

# pmos Instance M15 = spectreS device m15 mm15 net30 net30 in1 in1 \mpch_0p8
region=\triode w=40e-6 l=10e-6 m=1.0

# pmos Instance M49 = spectreS device m49 m49 out net30 in1 in1 \mpch_0p8
region=\triode w=40e-6 l=10e-6 m=1.0

# pmos Instance M50 = spectreS device m50 m50 in2 in2 in1 in1 \mpch_0p8 region=\tri-
ode w=40e-6 l=10e-6 m=1.0

# pmos Instance M16 = spectreS device m16 m16 net21 in2 in1 in1 \mpch_0p8
region=\triode w=40e-6 l=10e-6 m=1.0

```

```
# nmos Instance M58 = spectreS device m58 m58 net30 net21 net70 net70 \mnch_0p8  
region=\triode w=10e-6 l=10e-6 & m=1.0
```

```
# nmos Instance M57 = spectreS device m57 m57 net21 net21 net84 net84 \mnch_0p8  
region=\triode w=10e-6 l=10e-6 & m=1.0
```

```
# nmos Instance M3 = spectreS device m3 m3 net70 in3 0 0 \mnch_0p8 region=\triode  
w=10e-6 l=10e-6 m=1.0
```

```
# nmos Instance M1 = spectreS device m1 m1 net84 in1 0 0 \mnch_0p8 region=\triode  
w=10e-6 l=10e-6 m=1.0 # End of subcircuit definition. .ENDS &1
```

Appendix D

SPICE netlist of absolute value function circuit

```

il3 net1 0 isource type= dc dc=15e-6 m=1.0 il2 g1 net16 isource type= dc dc=0.0
m=1.0 v11 g1 0 vsource type= dc dc=5.0

```

```

m8 net7 net7 0 0 CMOSN region= triode w=2e-6 l=8e-6 as=+2.00000000E-12
+ad=+2.00000000E-12 ps=+6.00000000E-06 pd=+6.00000000E-06 nrd=+5.00000000E-
01 +nrs=+5.00000000E-01 m=1.0

```

```

m10 net10 net10 0 0 CMOSN region= triode w=2.2e-6 l=800e-9 +as=+2.20000000E-12
ad=+2.20000000E-12 ps=+6.40000000E-06 pd=+6.40000000E-06 +nrd=+4.54545455E-
01 nrs=+4.54545455E-01 m=1.0

```

```

m9 net15 net10 0 0 CMOSN region= triode w=2.2e-6 l=800e-9 +as=+2.20000000E-12
ad=+2.20000000E-12 ps=+6.40000000E-06 pd=+6.40000000E-06 +nrd=+4.54545455E-
01 nrs=+4.54545455E-01 m=1.0

```

```

m7 net16 net16 0 0 CMOSN region= triode w=2e-6 l=4e-6 as=+2.00000000E-12
+ad=+2.00000000E-12 ps=+6.00000000E-06 pd=+6.00000000E-06 nrd=+5.00000000E-
01 +nrs=+5.00000000E-01 m=1.0

```

```

m6 net24 net16 0 0 CMOSN region= triode w=2e-6 l=4e-6 as=+2.00000000E-12
+ad=+2.00000000E-12 ps=+6.00000000E-06 pd=+6.00000000E-06 nrd=+5.00000000E-
01 +nrs=+5.00000000E-01 m=1.0

```

```
m5 net24 net24 g1 g1 CMOSP region= triode w=1.2e-6 l=800e-9 +as=+1.20000000E-12
ad=+1.20000000E-12 ps=+4.40000000E-06 pd=+4.40000000E-06
+nrd=+8.33333333E-01 nrs=+8.33333333E-01 m=1.0
```

```
m4 net7 net24 g1 g1 CMOSP region= triode w=1.2e-6 l=800e-9 +as=+1.20000000E-12
ad=+1.20000000E-12 ps=+4.40000000E-06 pd=+4.40000000E-06 +nrd=+8.33333333E-01
nrs=+8.33333333E-01 m=1.0
```

```
m3 net7 net15 g1 g1 CMOSP region= triode w=1.2e-6 l=3e-6 as=+1.20000000E-12
+ad=+1.20000000E-12 ps=+4.40000000E-06 pd=+4.40000000E-06 nrd=+8.33333333E-01
+nrs=+8.33333333E-01 m=1.0
```

```
m2 net15 net15 g1 g1 CMOSP region= triode w=1.2e-6 l=3e-6 +as=+1.20000000E-12
ad=+1.20000000E-12 ps=+4.40000000E-06 pd=+4.40000000E-06 +nrd=+8.33333333E-01
nrs=+8.33333333E-01 m=1.0
```

```
m1 net1 net1 g1 g1 CMOSP region= triode w=1.2e-6 l=4e-6 as=+1.20000000E-12
+ad=+1.20000000E-12 ps=+4.40000000E-06 pd=+4.40000000E-06 nrd=+8.33333333E-01
+nrs=+8.33333333E-01 m=1.0
```

```
m0 net10 net1 g1 g1 CMOSP region= triode w=1.2e-6 l=4e-6 as=+1.20000000E-12
+ad=+1.20000000E-12 ps=+4.40000000E-06 pd=+4.40000000E-06 nrd=+8.33333333E-01
+nrs=+8.33333333E-01 m=1.0 .MODEL CMOSN NMOS LEVEL=3
```

```
VTO=+6.56600000E-01 PHI=0.700000 TOX=9.6000E-09 +XJ=0.200000U TPG=1
DELTA=6.9100E-01 LD=4.7290E-08 KP=1.9647E-04 UO=546.2 +THETA=2.6840E-01
RSH=3.5120E+01 GAMMA=0.5976 NSUB=1.3920E+17 NFS=5.9090E+11
+VMAX=2.0080E+05 ETA=3.7180E-02 KAPPA=2.8980E-02 CGDO=3.0515E-10
+CGSO=3.0515E-10 CGBO=4.0239E-10 CJ=5.62E-04 MJ=0.559 CJSW=5.00E-11
+MJSW=0.521 PB=0.99 XW=4.108E-07 XQC=0.5 XPART=0.5
```

```
.MODEL CMOSP PMOS LEVEL=3 PHI=0.700000 TOX=9.6000E-09 XJ=0.200000U
+TPG=-1.00000000E+00 VTO=-9.12200000E-01 DELTA=2.8750E-01 LD=3.5070E-08
+KP=4.8740E-05 UO=135.5 THETA=1.8070E-01 RSH=1.1000E-01
```

GAMMA=0.4673 +NSUB=8.5120E+16 NFS=6.5000E+11 VMAX=2.5420E+05
ETA=2.4500E-02 +KAPPA=7.9580E+00 CGDO=2.3922E-10 CGSO=2.3922E-10
CGBO=3.7579E-10 CJ=9.35E-04 +MJ=0.468 CJSW=2.89E-10 MJSW=0.505 PB=0.99
XW=3.622E-07 XQC=0.5 XPART=0.5 *

simulator lang=spectre

modelParameter info what=models where=rawfile

element info what=inst where=rawfile

outputParameter info what=output where=rawfile

simulator lang=spice

.TRAN 1.00000E-10 1.00000E-07 0.

simulator lang=spectre

finalTimeOP info what=oppoint where=rawfile

simulator lang=spice

.end

Appendix E

SPICE netlist of hysteresis circuit

```
# vpwl Instance V16 = spectreS device v16 v16 net24 net25 \source type=\pwl wave=[
0.0 0.0 50e-9 5.0 51e-9 5.0 & 100e-9 0.0\] delay=100e-9

# vdc Instance V17 = spectreS device v17 v17 in 0 \source type=\dc dc=vdc1

# vdc Instance V11 = spectreS device v11 v11 g1 0 \source type=\dc dc=5.0

# pfet3 Instance M3 = spectreS device m3 m3 0 out net4 g1 \CMOSP region=\triode
w=6e-6 l=1.2e-6 as=1e-6*6e-6 & ad=1e-6*6e-6 ps=2e-6+2.0*6e-6 pd=2e-6+2.0*6e-6
nrd=1e-6/6e-6 nrs=1e-6/6e-6 & m=1.0

# pfet3 Instance M2 = spectreS device m2 m2 out in net4 g1 \CMOSP region=\triode
w=14e-6 l=1.2e-6 as=1e-6*14e-6 & ad=1e-6*14e-6 ps=2e-6+2.0*14e-6 pd=2e-
6+2.0*14e-6 nrd=1e-6/14e-6 nrs=1e-6/& 14e-6 m=1.0

# pfet3 Instance M1 = spectreS device m1 m1 net4 in g1 g1 \CMOSP region=\triode
w=6e-6 l=1.2e-6 as=1e-6*6e-6 & ad=1e-6*6e-6 ps=2e-6+2.0*6e-6 pd=2e-6+2.0*6e-6
nrd=1e-6/6e-6 nrs=1e-6/6e-6 & m=1.0

# nfet3 Instance M5 = spectreS device m5 m5 g1 out net13 0 \CMOSN region=\triode
w=9.2e-6 l=1.2e-6 as=1e-6*9.2e-6 & ad=1e-6*9.2e-6 ps=2e-6+2.0*9.2e-6 pd=2e-
6+2.0*9.2e-6 nrd=1e-6/9.2e-6 & nrs=1e-6/9.2e-6 m=1.0
```

```
# nfet3 Instance M4 = spectreS device m4 m4 net13 in 0 0 \CMOSN region=\triode  
w=3e-6 l=1.2e-6 as=1e-6*3e-6 ad=1e-6*3e-6 ps=2e-6+2.0*3e-6 pd=2e-6+2.0*3e-6  
nrd=1e-6/3e-6 nrs=1e-6/3e-6 m=1.0 &
```

```
# nfet3 Instance M0 = spectreS device m0 m0 out in net13 0 \CMOSN region=\triode  
w=3.6e-6 l=1.2e-6 as=1e-6*3.6e-6 & ad=1e-6*3.6e-6 ps=2e-6+2.0*3.6e-6 pd=2e-  
6+2.0*3.6e-6 nrd=1e-6/3.6e-6 & nrs=1e-6/3.6e-6 m=1.0
```

Appendix F

SPICE netlist of current adder and subtractor

```
# vdc Instance V6 = spectreS device v6 v6 g1 0 \source type=\dc dc=5.0

# nfet3 Instance M12 = spectreS device m12 m12 g1 g1 net14 0 \CMOSN region=\triode
w=10e-6 l=2e-6 as=1e-6*10e-6 & ad=1e-6*10e-6 ps=2e-6+2.0*10e-6 pd=2e-6+2.0*10e-
6 nrd=1e-6/10e-6 nrs=1e-6/& 10e-6 m=1.0

# nfet3 Instance M8 = spectreS device m8 m8 net14 net6 0 0 \CMOSN region=\triode
w=4e-6 l=2e-6 as=1e-6*4e-6 ad=1e-6*4e-6 ps=2e-6+2.0*4e-6 pd=2e-6+2.0*4e-6
nrd=1e-6/4e-6 nrs=1e-6/4e-6 m=1.0 &

# nfet3 Instance M7 = spectreS device m7 m7 net6 net6 0 0 \CMOSN region=\triode
w=4e-6 l=2e-6 as=1e-6*4e-6 ad=1e-6*4e-6 ps=2e-6+2.0*4e-6 pd=2e-6+2.0*4e-6
nrd=1e-6/4e-6 nrs=1e-6/4e-6 m=1.0 &

# nfet3 Instance M4 = spectreS device m4 m4 net14 net15 0 0 \CMOSN region=\triode
w=4e-6 l=2e-6 as=1e-6*4e-6 & ad=1e-6*4e-6 ps=2e-6+2.0*4e-6 pd=2e-6+2.0*4e-6
nrd=1e-6/4e-6 nrs=1e-6/4e-6 & m=1.0

# nfet3 Instance M5 = spectreS device m5 m5 net15 net15 0 0 \CMOSN region=\triode
w=4e-6 l=2e-6 as=1e-6*4e-6 & ad=1e-6*4e-6 ps=2e-6+2.0*4e-6 pd=2e-6+2.0*4e-6
nrd=1e-6/4e-6 nrs=1e-6/4e-6 & m=1.0

# idc Instance I1 = spectreS device i1 i1 g1 net6 \source type=\dc dc=25e-6 m=1.0

# idc Instance I2 = spectreS device i2 i2 g1 net15 \source type=\dc dc=50e-6 m=1.0
```

*/SUBTRACTOR NETLIST

idc Instance I7 = spectreS device i7 i7 g1 net15 \source type=\dc dc=50e-6 m=1.0

idc Instance I6 = spectreS device i6 i6 g1 net24 \source type=\dc dc=100e-6 m=1.0

resistor Instance R5 = spectreS device r5 r5 net6 0 \resistor r=15e3 m=1.0

vdc Instance V4 = spectreS device v4 v4 g1 0 \source type=\dc dc=5.0

pfet3 Instance M3 = spectreS device m3 m3 net23 net23 g1 g1 \CMOSP region=\triode
w=10e-6 l=2e-6 as=1e-6*10e-6 & ad=1e-6*10e-6 ps=2e-6+2.0*10e-6 pd=2e-6+2.0*10e-
6 nrd=1e-6/10e-6 nrs=1e-6/& 10e-6 m=1.0

pfet3 Instance M2 = spectreS device m2 m2 net6 net23 g1 g1 \CMOSP region=\triode
w=10e-6 l=2e-6 as=1e-6*10e-6 & ad=1e-6*10e-6 ps=2e-6+2.0*10e-6 pd=2e-6+2.0*10e-
6 nrd=1e-6/10e-6 nrs=1e-6/& 10e-6 m=1.0

nfet3 Instance M11 = spectreS device m11 m11 net6 net15 0 0 \CMOSN region=\triode
w=4e-6 l=2e-6 as=1e-6*4e-6 & ad=1e-6*4e-6 ps=2e-6+2.0*4e-6 pd=2e-6+2.0*4e-6
nrd=1e-6/4e-6 nrs=1e-6/4e-6 & m=1.0

nfet3 Instance M10 = spectreS device m10 m10 net15 net15 0 0 \CMOSN region=\tri-
ode w=4e-6 l=2e-6 as=1e-6*4e-6 & ad=1e-6*4e-6 ps=2e-6+2.0*4e-6 pd=2e-6+2.0*4e-6
nrd=1e-6/4e-6 nrs=1e-6/4e-6 & m=1.0

nfet3 Instance M1 = spectreS device m1 m1 net24 net24 0 0 \CMOSN region=\triode
w=4e-6 l=2e-6 as=1e-6*4e-6 & ad=1e-6*4e-6 ps=2e-6+2.0*4e-6 pd=2e-6+2.0*4e-6
nrd=1e-6/4e-6 nrs=1e-6/4e-6 & m=1.0

nfet3 Instance M0 = spectreS device m0 m0 net23 net24 0 0 \CMOSN region=\triode
w=4e-6 l=2e-6 as=1e-6*4e-6 & ad=1e-6*4e-6 ps=2e-6+2.0*4e-6 pd=2e-6+2.0*4e-6
nrd=1e-6/4e-6 nrs=1e-6/4e-6 & m=1.0

Appendix G

SPICE netlist of a basic CNN cell

```
v48 in2 0 vsource type= dc dc=2.5 v47 in 0 vsource type= dc dc=2.5 v46 in1 0 vsource
type= dc dc=900e-3 v45 n5 0 vsource type= dc dc=5.0 r53 net118 n5 resistor r=10e3
m=1.0 r52 out net118 resistor r=1e3 m=1.0 r54 0 net118 resistor r=100e3 m=1.0
```

```
m45 out n4 n5 n5 CMOSP region= triode w=7.99999997980194e-6
+l=800.000009348878e-9 as=7.99999996803358e-12 ad=12.7999997753814e-12
+ps=9.99999974737875e-6 pd=19.2000006791204e-6 nrd=+1.25000000E-01
+nrs=+1.25000000E-01 m=1.0
```

```
m46 n5 n4 n4 n5 CMOSP region= triode w=7.99999997980194e-6
+l=800.000009348878e-9 as=12.7999997753814e-12 ad=7.99999996803358e-12
+ps=19.2000006791204e-6 pd=9.99999974737875e-6 nrd=+1.25000000E-01
+nrs=+1.25000000E-01 m=1.0
```

```
m47 n2 n1 n5 n5 CMOSP region= triode w=7.99999997980194e-6
+l=800.000009348878e-9 as=7.99999996803358e-12 ad=12.7999997753814e-12
+ps=9.99999974737875e-6 pd=19.2000006791204e-6 nrd=+1.25000000E-01
+nrs=+1.25000000E-01 m=1.0
```

```
m48 n5 n1 n1 n5 CMOSP region= triode w=7.99999997980194e-6
+l=800.000009348878e-9 as=12.7999997753814e-12 ad=7.99999996803358e-12
+ps=19.2000006791204e-6 pd=9.99999974737875e-6 nrd=+1.25000000E-01
+nrs=+1.25000000E-01 m=1.0
```

m49 n2 0 in n5 CMOS region= triode w=800.000009348878e-9
 +l=27.5999991572462e-6 as=4.88000004988942e-12 ad=4.48000013822392e-12
 +ps=10.2000003607827e-6 pd=9.20000002224697e-6 nrd=+1.24999999E+00
 +nrs=+1.24999999E+00 m=1.0

m50 in n5 n2 0 CMOSN region= triode w=800.000009348878e-9
 +l=24.9999993684469e-6 as=4.39999989568229e-12 ad=4.71999999848705e-12
 +ps=9.0000003183377e-6 pd=9.80000004346948e-6 nrd=+1.24999999E+00
 +nrs=+1.24999999E+00 m=1.0

m51 0 n2 0 0 CMOSN region= triode w=21.1999995372025e-6
 +l=9.99999974737875e-6 as=+2.11999995E-11 ad=69.959996129576e-12
 +ps=+4.43999991E-05 pd=91.3999974727631e-6 nrd=+4.71698124E-02
 +nrs=+4.71698124E-02 m=1.0

m52 out in2 n3 0 CMOSN region= triode w=9.20000002224697e-6
 +l=800.000009348878e-9 as=9.20000013671096e-12 ad=14.7199995248481e-12
 +ps=11.1999997898238e-6 pd=21.6000007640105e-6 nrd=+1.08695652E-01
 +nrs=+1.08695652E-01 m=1.0

m53 n3 n2 n4 0 CMOSN region= triode w=9.20000002224697e-6
 +l=800.000009348878e-9 as=14.7199995248481e-12 ad=9.20000013671096e-12
 +ps=21.6000007640105e-6 pd=11.1999997898238e-6 nrd=+1.08695652E-01
 +nrs=+1.08695652E-01 m=1.0

m54 n3 in1 0 0 CMOSN region= triode w=5.200000032346e-6 +l=800.000009348878e-9
 as=8.32000007083833e-12 ad=8.8399999126354e-12 +ps=13.5999998747138e-6
 pd=13.7999995786231e-6 nrd=+1.92307691E-01 +nrs=+1.92307691E-01 m=1.0

m55 n2 in2 n0 0 CMOSN region= triode w=9.20000002224697e-6
 +l=800.000009348878e-9 as=9.20000013671096e-12 ad=14.7199995248481e-12
 +ps=11.1999997898238e-6 pd=21.6000007640105e-6 nrd=+1.08695652E-01
 +nrs=+1.08695652E-01 m=1.0

```

m56 n0 n2 n1 0 CMOSN region= triode w=9.20000002224697e-6
+l=800.000009348878e-9 as=14.7199995248481e-12 ad=9.20000013671096e-12
+ps=21.6000007640105e-6 pd=11.1999997898238e-6 nrd=+1.08695652E-01
+nrs=+1.08695652E-01 m=1.0

m57 n0 in1 0 0 CMOSN region= triode w=5.200000032346e-6 +l=800.000009348878e-
9 as=8.32000007083833e-12 ad=8.8399999126354e-12 +ps=13.5999998747138e-6
pd=13.7999995786231e-6 nrd=+1.92307691E-01 +nrs=+1.92307691E-01 m=1.0

.MODEL CMOSP PMOS LEVEL=3 PHI=0.700000 TOX=9.6000E-09 XJ=0.200000U
+TPG=-1.00000000E+00 VTO=-9.12200000E-01 DELTA=2.8750E-01 LD=3.5070E-08
+KP=4.8740E-05 UO=135.5 THETA=1.8070E-01 RSH=1.1000E-01
GAMMA=0.4673 +NSUB=8.5120E+16 NFS=6.5000E+11 VMAX=2.5420E+05
ETA=2.4500E-02 +KAPPA=7.9580E+00 CGDO=2.3922E-10 CGSO=2.3922E-10
CGBO=3.7579E-10 CJ=9.35E-04 +MJ=0.468 CJSW=2.89E-10 MJSW=0.505 PB=0.99
XW=3.622E-07 XQC=0.5 XPART=0.5

.MODEL CMOSN NMOS LEVEL=3 VTO=+6.56600000E-01 PHI=0.700000
TOX=9.6000E-09 +XJ=0.200000U TPG=1 DELTA=6.9100E-01 LD=4.7290E-08
KP=1.9647E-04 UO=546.2 +THETA=2.6840E-01 RSH=3.5120E+01
GAMMA=0.5976 NSUB=1.3920E+17 NFS=5.9090E+11 +VMAX=2.0080E+05
ETA=3.7180E-02 KAPPA=2.8980E-02 CGDO=3.0515E-10 +CGSO=3.0515E-10
CGBO=4.0239E-10 CJ=5.62E-04 MJ=0.559 CJSW=5.00E-11 +MJSW=0.521 PB=0.99
XW=4.108E-07 XQC=0.5 XPART=0.5

.TRAN 1.00000E-10 2.00000E-07 0.

simulator lang=spectre

simulator lang=spice

.end

```

Appendix H

SPICE netlist of Rule 2 verification

```
# capacitor Instance C50 = spectreS device c50 c50 net11 0 \capacitor c=100e-15 m=1.0
ic=0.0

# neuronR Instance I47 = spectreS device xi47 xi47 net56 net55 net54 net83 net5
cell_neuronR_schematic

# neuronR1 Instance I46 = spectreS device xi46 xi46 net56 net55 net54 net68 net57
cell_neuronR1_schematic

# inverter1 Instance I45 = spectreS device xi45 xi45 net11 net10 cell_inverter1_schematic

# resistor Instance R37 = spectreS device r37 r37 net57 net1 \resistor r=10e3 m=1.0

# resistor Instance R38 = spectreS device r38 r38 net5 net1 \resistor r=10e3 m=1.0

# resistor Instance R27 = spectreS device r27 r27 net1 net20 \resistor r=10e3 m=1.0

# hyteresis Instance I25 = spectreS device xi25 xi25 net10 net44 cell_hyteresis_schematic

# nfet3 Instance M49 = spectreS device m49 m49 net83 net44 0 0 \CMOSN region=\tri-
ode w=2e-6 l=2e-6 as=1e-6*2e-6 & ad=1e-6*2e-6 ps=2e-6+2.0*2e-6 pd=2e-6+2.0*2e-6
nrd=1e-6/2e-6 nrs=1e-6/2e-6 & m=1.0
```

```
# nfet3 Instance M33 = spectreS device m33 m33 g1 net44 net68 0 \CMOSN region=\triode
w=2e-6 l=2e-6 as=1e-6*2e-6 & ad=1e-6*2e-6 ps=2e-6+2.0*2e-6 pd=2e-6+2.0*2e-6
nrd=1e-6/2e-6 nrs=1e-6/2e-6 & m=1.0
```

```
# nfet3 Instance M22 = spectreS device m22 m22 net11 net11 0 0 \CMOSN region=\triode
w=2e-6 l=20e-6 as=1e-6*2e-6 & ad=1e-6*2e-6 ps=2e-6+2.0*2e-6 pd=2e-6+2.0*2e-6
nrd=1e-6/2e-6 nrs=1e-6/2e-6 & m=1.0
```

```
# nfet3 Instance M42 = spectreS device m42 m42 net14 net14 0 0 \CMOSN region=\triode
w=4e-6 l=4e-6 as=1e-6*4e-6 & ad=1e-6*4e-6 ps=2e-6+2.0*4e-6 pd=2e-6+2.0*4e-6
nrd=1e-6/4e-6 nrs=1e-6/4e-6 & m=1.0
```

```
# nfet3 Instance M13 = spectreS device m13 m13 net19 net14 0 0 \CMOSN region=\triode
w=4e-6 l=4e-6 as=1e-6*4e-6 & ad=1e-6*4e-6 ps=2e-6+2.0*4e-6 pd=2e-6+2.0*4e-6
nrd=1e-6/4e-6 nrs=1e-6/4e-6 & m=1.0
```

```
# nfet3 Instance M41 = spectreS device m41 m41 net20 net20 0 0 \CMOSN region=\triode
w=2e-6 l=10e-6 as=1e-6*2e-6 & ad=1e-6*2e-6 ps=2e-6+2.0*2e-6 pd=2e-6+2.0*2e-6
nrd=1e-6/2e-6 nrs=1e-6/2e-6 & m=1.0
```

```
# nfet3 Instance M10 = spectreS device m10 m10 net28 net20 0 0 \CMOSN region=\triode
w=2e-6 l=10e-6 as=1e-6*2e-6 & ad=1e-6*2e-6 ps=2e-6+2.0*2e-6 pd=2e-6+2.0*2e-6
nrd=1e-6/2e-6 nrs=1e-6/2e-6 & m=1.0
```

```
# pfet3 Instance M18 = spectreS device m18 m18 net28 net28 g1 g1 \CMOSP
region=\triode w=6e-6 l=4e-6 as=1e-6*6e-6 & ad=1e-6*6e-6 ps=2e-6+2.0*6e-6 pd=2e-6+2.0*6e-6
nrd=1e-6/6e-6 nrs=1e-6/6e-6 & m=1.0
```

```
# pfet3 Instance M17 = spectreS device m17 m17 net11 net28 g1 g1 \CMOSP
region=\triode w=6e-6 l=4e-6 as=1e-6*6e-6 & ad=1e-6*6e-6 ps=2e-6+2.0*6e-6 pd=2e-6+2.0*6e-6
nrd=1e-6/6e-6 nrs=1e-6/6e-6 & m=1.0
```



```

# pfet3 Instance M16 = spectreS device m16 m16 net11 net19 g1 g1 \CMOSP
region=\triode w=6e-6 l=4e-6 as=1e-6*6e-6 & ad=1e-6*6e-6 ps=2e-6+2.0*6e-6 pd=2e-
6+2.0*6e-6 nrd=1e-6/6e-6 nrs=1e-6/6e-6 & m=1.0

# pfet3 Instance M15 = spectreS device m15 m15 net19 net19 g1 g1 \CMOSP
region=\triode w=6e-6 l=4e-6 as=1e-6*6e-6 & ad=1e-6*6e-6 ps=2e-6+2.0*6e-6 pd=2e-
6+2.0*6e-6 nrd=1e-6/6e-6 nrs=1e-6/6e-6 & m=1.0

# pfet3 Instance M11 = spectreS device m11 m11 net20 net20 g1 g1 \CMOSP
region=\triode w=2e-6 l=10e-6 as=1e-6*2e-6 & ad=1e-6*2e-6 ps=2e-6+2.0*2e-6 pd=2e-
6+2.0*2e-6 nrd=1e-6/2e-6 nrs=1e-6/2e-6 & m=1.0

# pfet3 Instance M9 = spectreS device m9 m9 net14 net20 g1 g1 \CMOSP region=\triode
w=2e-6 l=10e-6 as=1e-6*2e-6 & ad=1e-6*2e-6 ps=2e-6+2.0*2e-6 pd=2e-6+2.0*2e-6
nrd=1e-6/2e-6 nrs=1e-6/2e-6 & m=1.0

# vdc Instance V7 = spectreS device v7 v7 net54 0 \vsource type=\dc dc=2.5

# vdc Instance V4 = spectreS device v4 v4 g1 0 \vsource type=\dc dc=5.0

# vdc Instance V3 = spectreS device v3 v3 net55 0 \vsource type=\dc dc=1.0

# vdc Instance V2 = spectreS device v2 v2 net56 0 \vsource type=\dc dc=2.5

*/ CELL N1

# terminal mapping: IN = IN #          IN1 = IN1 #          IN2 = IN2 #
IN3 = IN3 #          OUT = OUT

.SUBCKT &1 IN IN1 IN2 IN3 OUT

# capacitor Instance C60 = spectreS device c60 c60 in3 0 \capacitor c=10e-15 m=1.0
ic=500e-3

```

```

# nfet3 Instance M55 = spectreS device m55 m55 in3 in2 net63 0 \CMOSN region=\triode
w=9.2e-6 l=800e-9 as=1e-6*9.2e-6 & 6 ad=1e-6*9.2e-6 ps=2e-6+2.0*9.2e-6 pd=2e-
6+2.0*9.2e-6 nrd=1e-6/9.2e-6 & nrs=1e-6/9.2e-6 m=1.0

# nfet3 Instance M42 = spectreS device m42 m42 0 in3 0 0 \CMOSN region=\triode
w=21.2e-6 l=10e-6 as=1e-6*21.2e-6 & ad=1e-6*21.2e-6 ps=2e-6+2.0*21.2e-6 pd=2e-
6+2.0*21.2e-6 nrd=1e-6/21.2e-6 & nrs=1e-6/21.2e-6 m=1.0

# nfet3 Instance M21 = spectreS device m21 m21 net54 in3 net57 0 \CMOSN
region=\triode w=9.2e-6 l=800e-9 as=1e-6*& 9.2e-6 ad=1e-6*9.2e-6 ps=2e-6+2.0*9.2e-6
pd=2e-6+2.0*9.2e-6 nrd=1e-6/9.2e-6 & 6 nrs=1e-6/9.2e-6 m=1.0

# nfet3 Instance M17 = spectreS device m17 m17 net57 in1 0 0 \CMOSN region=\triode
w=5.2e-6 l=800e-9 as=1e-6*5.2e-6 & ad=1e-6*5.2e-6 ps=2e-6+2.0*5.2e-6 pd=2e-
6+2.0*5.2e-6 nrd=1e-6/5.2e-6 & nrs=1e-6/5.2e-6 m=1.0

# nfet3 Instance M57 = spectreS device m57 m57 out in2 net57 0 \CMOSN region=\triode
w=9.2e-6 l=800e-9 as=1e-6*9.2e-6 & 6 ad=1e-6*9.2e-6 ps=2e-6+2.0*9.2e-6 pd=2e-
6+2.0*9.2e-6 nrd=1e-6/9.2e-6 & nrs=1e-6/9.2e-6 m=1.0

# nfet3 Instance M6 = spectreS device m6 m6 net63 in1 0 0 \CMOSN region=\triode
w=5.2e-6 l=800e-9 as=1e-6*5.2e-6 & ad=1e-6*5.2e-6 ps=2e-6+2.0*5.2e-6 pd=2e-
6+2.0*5.2e-6 nrd=1e-6/5.2e-6 & nrs=1e-6/5.2e-6 m=1.0

# nfet3 Instance M5 = spectreS device m5 m5 net66 in3 net63 0 \CMOSN region=\triode
w=9.2e-6 l=800e-9 as=1e-6*& 9.2e-6 ad=1e-6*9.2e-6 ps=2e-6+2.0*9.2e-6 pd=2e-
6+2.0*9.2e-6 nrd=1e-6/9.2e-6 & 6 nrs=1e-6/9.2e-6 m=1.0

# pfet3 Instance M19 = spectreS device m19 m19 net54 net54 g1 g1 \CMOSP
region=\triode w=8e-6 l=800e-9 as=1e-6*8e-6 & ad=1e-6*8e-6 ps=2e-6+2.0*8e-6 pd=2e-
6+2.0*8e-6 nrd=1e-6/8e-6 nrs=1e-6/8e-6 & m=1.0

```

```
# pfet3 Instance M20 = spectreS device m20 m20 out net54 g1 g1 \CMOSP region=\triode
w=8e-6 l=800e-9 as=1e-6*8e-6 & ad=1e-6*8e-6 ps=2e-6+2.0*8e-6 pd=2e-6+2.0*8e-6
nrd=1e-6/8e-6 nrs=1e-6/8e-6 & m=1.0
```

```
# pfet3 Instance M4 = spectreS device m4 m4 net66 net66 g1 g1 \CMOSP region=\triode
w=8e-6 l=800e-9 as=1e-6*8e-6 & ad=1e-6*8e-6 ps=2e-6+2.0*8e-6 pd=2e-6+2.0*8e-6
nrd=1e-6/8e-6 nrs=1e-6/8e-6 & m=1.0
```

```
# pfet3 Instance M2 = spectreS device m2 m2 in3 net66 g1 g1 \CMOSP region=\triode
w=8e-6 l=800e-9 as=1e-6*8e-6 & ad=1e-6*8e-6 ps=2e-6+2.0*8e-6 pd=2e-6+2.0*8e-6
nrd=1e-6/8e-6 nrs=1e-6/8e-6 & m=1.0
```

```
# pfet Instance M59 = spectreS device m59 m59 in3 0 in g1 \CMOSP region=\triode
w=800e-9 l=27.2e-6 as=1e-6*800e-9 & ad=1e-6*800e-9 ps=2e-6+2.0*800e-9 pd=2e-6+2.0*800e-9
nrd=1e-6/800e-9 & nrs=1e-6/800e-9 m=1.0
```

```
# nfet Instance M0 = spectreS device m0 m0 in3 g1 in 0 \CMOSN region=\triode
w=800e-9 l=25e-6 as=1e-6*800e-9 & ad=1e-6*800e-9 ps=2e-6+2.0*800e-9 pd=2e-6+2.0*800e-9
nrd=1e-6/800e-9 & nrs=1e-6/800e-9 m=1.0 # End of subcircuit definition.
.ENDS &1
```

```
*/ CELL N2
```

```
# terminal mapping: IN = IN #          IN1 = IN1 #          IN2 = IN2 #
IN3 = IN3 #          OUT = OUT
```

```
.SUBCKT &1 IN IN1 IN2 IN3 OUT
```

```
# capacitor Instance C60 = spectreS device c60 c60 in3 0 \capacitor c=10e-15 m=1.0
ic=4.0
```

```
# nfet3 Instance M55 = spectreS device m55 m55 in3 in2 net63 0 \CMOSN region=\triode
w=9.2e-6 l=800e-9 as=1e-6*9.2e-6 & ad=1e-6*9.2e-6 ps=2e-6+2.0*9.2e-6 pd=2e-6+2.0*9.2e-6
nrd=1e-6/9.2e-6 & nrs=1e-6/9.2e-6 m=1.0
```

```
# nfet3 Instance M42 = spectreS device m42 m42 0 in3 0 0 \CMOSN region=\triode
w=21.2e-6 l=10e-6 as=1e-6*21.2e-6 & ad=1e-6*21.2e-6 ps=2e-6+2.0*21.2e-6 pd=2e-6+2.0*21.2e-6
nrd=1e-6/21.2e-6 & nrs=1e-6/21.2e-6 m=1.0
```

```
# nfet3 Instance M21 = spectreS device m21 m21 net54 in3 net57 0 \CMOSN
region=\triode w=9.2e-6 l=800e-9 as=1e-6*9.2e-6 ad=1e-6*9.2e-6 ps=2e-6+2.0*9.2e-6
pd=2e-6+2.0*9.2e-6 nrd=1e-6/9.2e-6 & nrs=1e-6/9.2e-6 m=1.0
```

```
# nfet3 Instance M17 = spectreS device m17 m17 net57 in1 0 0 \CMOSN region=\triode
w=5.2e-6 l=800e-9 as=1e-6*5.2e-6 & ad=1e-6*5.2e-6 ps=2e-6+2.0*5.2e-6 pd=2e-6+2.0*5.2e-6
nrd=1e-6/5.2e-6 & nrs=1e-6/5.2e-6 m=1.0
```

```
# nfet3 Instance M57 = spectreS device m57 m57 out in2 net57 0 \CMOSN region=\triode
w=9.2e-6 l=800e-9 as=1e-6*9.2e-6 & ad=1e-6*9.2e-6 ps=2e-6+2.0*9.2e-6 pd=2e-6+2.0*9.2e-6
nrd=1e-6/9.2e-6 & nrs=1e-6/9.2e-6 m=1.0
```

```
# nfet3 Instance M6 = spectreS device m6 m6 net63 in1 0 0 \CMOSN region=\triode
w=5.2e-6 l=800e-9 as=1e-6*5.2e-6 & ad=1e-6*5.2e-6 ps=2e-6+2.0*5.2e-6 pd=2e-6+2.0*5.2e-6
nrd=1e-6/5.2e-6 & nrs=1e-6/5.2e-6 m=1.0
```

```
# nfet3 Instance M5 = spectreS device m5 m5 net66 in3 net63 0 \CMOSN region=\triode
w=9.2e-6 l=800e-9 as=1e-6*9.2e-6 ad=1e-6*9.2e-6 ps=2e-6+2.0*9.2e-6 pd=2e-6+2.0*9.2e-6
nrd=1e-6/9.2e-6 & nrs=1e-6/9.2e-6 m=1.0
```

```
# pfet3 Instance M19 = spectreS device m19 m19 net54 net54 g1 g1 \CMOSP
region=\triode w=8e-6 l=800e-9 as=1e-6*8e-6 & ad=1e-6*8e-6 ps=2e-6+2.0*8e-6 pd=2e-6+2.0*8e-6
nrd=1e-6/8e-6 nrs=1e-6/8e-6 & m=1.0
```

```
# pfet3 Instance M20 = spectreS device m20 m20 out net54 g1 g1 \CMOSP region=\triode
w=8e-6 l=800e-9 as=1e-6*8e-6 & ad=1e-6*8e-6 ps=2e-6+2.0*8e-6 pd=2e-6+2.0*8e-6
nrd=1e-6/8e-6 nrs=1e-6/8e-6 & m=1.0
```

```
# pfet3 Instance M4 = spectreS device m4 m4 net66 net66 g1 g1 \CMOSP region=\triode
w=8e-6 l=800e-9 as=1e-6*8e-6 & ad=1e-6*8e-6 ps=2e-6+2.0*8e-6 pd=2e-6+2.0*8e-6
nrd=1e-6/8e-6 nrs=1e-6/8e-6 & m=1.0
```

```
# pfet3 Instance M2 = spectreS device m2 m2 in3 net66 g1 g1 \CMOSP region=\triode
w=8e-6 l=800e-9 as=1e-6*8e-6 & ad=1e-6*8e-6 ps=2e-6+2.0*8e-6 pd=2e-6+2.0*8e-6
nrd=1e-6/8e-6 nrs=1e-6/8e-6 & m=1.0
```

```
# pfet Instance M59 = spectreS device m59 m59 in3 0 in g1 \CMOSP region=\triode
w=800e-9 l=27.2e-6 as=1e-6*800e-9 & ad=1e-6*800e-9 ps=2e-6+2.0*800e-9 pd=2e-6+2.0*800e-9
nrd=1e-6/800e-9 & nrs=1e-6/800e-9 m=1.0
```

```
# nfet Instance M0 = spectreS device m0 m0 in3 g1 in 0 \CMOSN region=\triode
w=800e-9 l=25e-6 as=1e-6*800e-9 & ad=1e-6*800e-9 ps=2e-6+2.0*800e-9 pd=2e-6+2.0*800e-9
nrd=1e-6/800e-9 & nrs=1e-6/800e-9 m=1.0 # End of subcircuit definition.
.ENDS &1 # terminal mapping: IN = IN # OUT = OUT
```

```
*/ INVERTER
```

```
.SUBCKT &1 IN OUT
```

```
# nfet3 Instance M3 = spectreS device m3 m3 out in 0 0 \CMOSN region=\triode w=8e-6
l=800e-9 as=1e-6*8e-6 ad=1e-6*& 8e-6 ps=2e-6+2.0*8e-6 pd=2e-6+2.0*8e-6 nrd=1e-6/8e-6
nrs=1e-6/8e-6 m=1.0
```

```
# pfet3 Instance M2 = spectreS device m2 m2 out in g1 g1 \CMOSP region=\triode
w=2e-6 l=800e-9 as=1e-6*2e-6 ad=1e-& 6*2e-6 ps=2e-6+2.0*2e-6 pd=2e-6+2.0*2e-6
nrd=1e-6/2e-6 nrs=1e-6/2e-6 m=1.0 & # End of subcircuit definition. .ENDS &1 # terminal mapping: IN = IN # OUT = OUT
```

***/ HYSTERISIS CIRCUIT**
.SUBCKT &1 IN OUT
capacitor Instance C12 = spectreS device c12 c12 out 0 \capacitor c=1e-12 m=1.0 ic=0.0
capacitor Instance C13 = spectreS device c13 c13 in 0 \capacitor c=1e-12 m=1.0 ic=5.0
**# pfet3 Instance M3 = spectreS device m3 m3 0 out net4 g1 \CMOSP region=\triode
w=14e-6 l=6e-6 as=1e-6*14e-6 & ad=1e-6*14e-6 ps=2e-6+2.0*14e-6 pd=2e-6+2.0*14e-
6 nrd=1e-6/14e-6 nrs=1e-6/& 14e-6 m=1.0**
**# pfet3 Instance M2 = spectreS device m2 m2 out in net4 g1 \CMOSP region=\triode
w=14e-6 l=6e-6 as=1e-6*14e-6 & ad=1e-6*14e-6 ps=2e-6+2.0*14e-6 pd=2e-6+2.0*14e-
6 nrd=1e-6/14e-6 nrs=1e-6/& 14e-6 m=1.0**
**# pfet3 Instance M1 = spectreS device m1 m1 net4 in g1 g1 \CMOSP region=\triode
w=14e-6 l=6e-6 as=1e-6*14e-6 & ad=1e-6*14e-6 ps=2e-6+2.0*14e-6 pd=2e-6+2.0*14e-
6 nrd=1e-6/14e-6 nrs=1e-6/& 14e-6 m=1.0**
**# nfet3 Instance M5 = spectreS device m5 m5 g1 out net13 0 \CMOSN region=\triode
w=44e-6 l=6e-6 as=1e-6*44e-6 & ad=1e-6*44e-6 ps=2e-6+2.0*44e-6 pd=2e-6+2.0*44e-
6 nrd=1e-6/44e-6 nrs=1e-6/& 44e-6 m=1.0**
**# nfet3 Instance M4 = spectreS device m4 m4 net13 in 0 0 \CMOSN region=\triode
w=6e-6 l=6e-6 as=1e-6*6e-6 ad=1e-6*& 6e-6 ps=2e-6+2.0*6e-6 pd=2e-6+2.0*6e-6
nrd=1e-6/6e-6 nrs=1e-6/6e-6 m=1.0**
**# nfet3 Instance M0 = spectreS device m0 m0 out in net13 0 \CMOSN region=\triode
w=18e-6 l=6e-6 as=1e-6*18e-6 & ad=1e-6*18e-6 ps=2e-6+2.0*18e-6 pd=2e-6+2.0*18e-
6 nrd=1e-6/18e-6 nrs=1e-6/& 18e-6 m=1.0**
End of subcircuit definition. .ENDS &1

Vita Auctoris

Anu Mathew was in born Kerala, India on March 31, 1992. He finished his high school in St.Jude High School, Quilon, Kerala and B.E in Computer Science and Engineering from Marathawada University, India in 1993. He completed his M.A.Sc in Electrical Engineering at the University of Windsor in December 1997. He is currently employed at NORTEL Ottawa. His research interests include Analog/Digital Circuit Design and Cellular Neural Networks.